



Universidad Carlos III de Madrid

Departamento de Ingeniería Telemática

Trabajo de Fin de Grado

Grado en Ingeniería en Tecnologías de Telecomunicación.

# **SISTEMA DE REALIDAD AUMENTADA PARA CONTROL DOMÓTICO**

---

Autor: Eugenio Rubio Drosdov

Tutor: Daniel Díaz Sánchez

*A Tatiana, Carmen y Aurora: ellas lo han hecho posible.*

## Agradecimientos

Después de mucho tiempo y muchas vivencias, se cierra una etapa y se abre otra.

Esta carrera no ha sido fácil. Y digo carrera atendiendo al hecho de que tiene una línea de salida y una línea de llegada. Tiene pendientes difíciles de superar y alguna que otra llanura. Tiene competidores, orientadores, entrenadores incluso... Pero, desde luego, lo más importante no es correrla o andarla, sino superarla. Muchos se han retirado de ella, muchos han llegado antes que yo y otros llegarán más tarde, pero lo importante es que yo estoy aquí.

He aprendido a pensar, he crecido como persona, soy totalmente diferente a aquél adolescente que entró queriéndose comer el mundo y que en el primer año se dió de frente con la verdad, como tantos otros. Ahora sigo queriendo comerme el mundo y sé cómo puedo hacerlo.

Pero esta sensación de victoria no llega sólo por un trabajo personal, aunque sea un gran porcentaje de este, sino que también llega por la ayuda que te ofrecen.

La infancia que he tenido se caracteriza por una familia volcada en darme una educación, por una madre, una abuela y una tía abuela que lo han dado todo para que yo llegase a ser alguien. Los sacrificios que han hecho ni si quiera puedo acercarme a recompensárselos... Después llegó la adolescencia, aquella etapa rebelde en la que no te das cuenta de que tu familia está siempre a tu lado y que incluso piensas que te quieren incoordinar, pero ellas siguen estando ahí. Más tarde, la universidad. Etapa de descubrimiento personal, de crecimiento, de ser adulto y formar parte activa de una sociedad a la espera de que aportes algo. De nuevo, tu familia lo da todo por ti. Lo único que puedo decir es “gracias de todo corazón por ayudarme a llegar hasta aquí”.

Por supuesto, en esta última etapa, la universidad, la gente con la que te encuentras es importante. Profesores, compañeros hacen de esta vivencia una etapa única, llena de alegrías y penas. Examen tras examen te vas descubriendo. Prueba tras prueba tu cabeza desarrolla habilidades que te conducen a optar por un camino dentro de tu campo. Yo sé qué quiero ser de mayor. Y lo sé gracias a los profesores que me han marcado como Florina Almenares, Andrés Marín y por supuesto Daniel Díaz Sánchez, tutor de este trabajo. Ellos me han enseñado una rama de la ingeniería de telecomunicaciones atrac-

tiva. Me han hecho conocerla, entenderla y vivirla.

Quiero dedicar un párrafo a mi tutor, Dani, para agradecerle de forma más extendida su apoyo y dirección en este proyecto. Desde el día en el que se lo propuse, sus ganas e ideas no paraban de fluir. Reunión tras reunión veía más claro que elegirle como tutor había sido una gran idea. A lo largo del desarrollo hemos congeniado y pensado como llevarlo a cabo, desarrollando una confianza laboral que ha hecho más sencilla y fluida la finalización de esta etapa. Gracias Dani.

Y por supuesto resaltar la forma en que amigos y conocidos me han apoyado, dando ánimos y entendiendo en muchas ocasiones mi ausencia social por el desarrollo de este prototipo.

## Resumen

La Realidad Aumentada (RA) es un concepto que se considera novedoso, ya que mezcla en nuestros dispositivos el mundo real con un mundo ficticio que querramos crear en tiempo real, haciéndolo visual y de fácil control, permitiendo que interacciones complejas sean fáciles de realizarse, integrando capas de información gracias a las cámaras que prácticamente todos los dispositivos móviles o tablets han incorporado.

La capacidad de nuestros teléfonos móviles o tablets crece de forma exponencial, por lo que disponen de los recursos necesarios para este tipo de Aplicaciones que mezclan, tanto el control de la cámara, como el de una extensa base de datos, así como un motor gráfico, que muestra el escenario que queremos superponer a lo que vemos.

Por otro lado, el control domótico cada vez está formando parte más natural en el diseño y control de nuestros hogares, abriéndose un gran abanico de posibilidades que van, desde contactar nuestras casa con un servicio de seguridad, hasta hacer la compra a través del frigorífico de forma automática.

Este Trabajo de Fin de Grado describe una aplicación sencilla en entorno Android que une estos conceptos, haciendo el control del hogar más sencillo y visual, utilizando herramientas de Realidad Aumentada para detectar elementos configurables de un ambiente automatizado mediante un interfaz claro e intuitivo, adaptado a usuarios no profesionales.

## Abstract

Augmented Reality (AR) is an innovative concept, that provides the possibility to combine in real time, in the devices designed to be used for everyone, a real world with the one we want to create, as the information about the surrounding real world of the user becomes interactive and digitally manipulable. An augmented reality system makes this procedure visual and easy to control, as the system allows complex interactions to be realized in a very simple way, settling the information layers through the use of cameras that each cell phone or tablet has.

The capacity of tablets and cell phones normally used grows exponentially. Due to this, these devices have the resources needed for this kind of applications. They combine the use of the camera and the use of a large data base, just like a graphic engine that shows the scene we want to overlap.

On the other hand, the domotic control has become a natural part of the design and the control of our homes, displaying a lot of options, which include having the houses connected to a security service or doing shopping using a refrigerator.

This Graduate Project describes a simple Android app aimed to do home control visual and easy. The Augmented Reality tools have been used to design elements detection and to make easier the interaction operations by using a simple and intuitive interface adapted for the non professional users.

# Synopsis

## Introduction

The entry of smartphones and mobile devices in the market gives a new meaning to the term “be connected”. The users not only want to be contacted with other people, but also desire to do it 24 hours a day: we want to be continuously connected.

Imagine that we are able to connect each element of our houses to a small network, to which we are usually connected. We would control any device, raise or lower the shutters, feed our pets. . . But, what if we get to do these operations simple, in addition to doing it possible?

This Graduate Paper is an attempt to go a step further in order to propose an operative solution that allows control any desired element using some clear and simple options of the interface.

The Augmented Reality boasts about its ability to interact with the environment using the camera of the device involved to control it.

Therefore, the tags we had before us were the following: to recognize the domestic environment and to interact with each element. That’s it: to be able to recognize what kind of element of the environment is and which operation we want to apply.

The interaction is to be held by an attractive, clear and intuitive option, which is the direct use of the image that the implied device offers through its camera.

The camera installed allows to observe the real world as well as to select the object or some objects on which the user acts with the purpose to modify their working mode.

In order to analyze the environment and specify what we can do with these objects, and, specially, in order to carry out different actions, we need to:

- Develop a system for object recognition. What is around us?
- Develop a system of control over the actions as regard different objects. What can we do with these items?

- Develop an action system over these ones. How we do it?

## State of the art

The history of domotics starts at 70s, when buildings begins to be automatized, but it was not really until the 80s when it starts to resemble the concept that we have now.

Initially, the unique domotics installation that could be done consisted in the use of sensors and actuators that links to an automaton using a centralized architecture. It was inflexible, which possible extensions were expensive and hard to be done.

There are four main types of architectures:

1. Centralized.
2. Decentralized.
3. Distributed.
4. Hybrid.

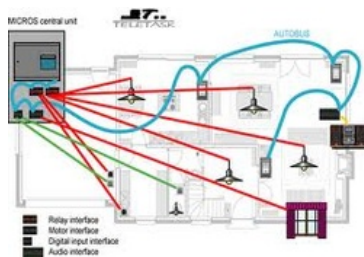


Figura 1: Centralized

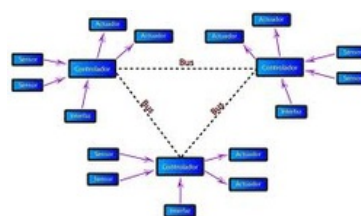


Figura 2: Decentralized



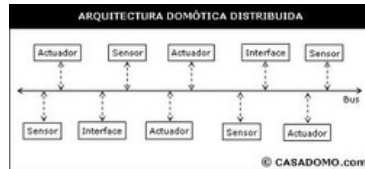


Figura 3: Distributed



Figura 4: Hybrid

All this exchange of information happens because of X10 or KNX / EIB (Europe) and Lonworks(U.S.A.) protocols.

The X-10 protocol is a standard for establishing communication between different compatible devices.

The European Installation Bus, or EIB, is the next step from the point of view of scalability.

Unlike X10, EIB need an added wiring. This extra wiring is a parallel control bus of building wires, which all parts are connected to, resulting in a no needed central control unit such as a computer or server.

Augmented Reality is a field relatively recent.

First of all we must differentiate between AR and Virtual Reality (VR). The major difference is the uses of a camera to present that Reality. In the case of the VR everything is available in digital format, creating the environment in this format, while the AR overlaps a new reality to our own real world seen through camera's view.

When talking about AR should differentiate between two types of options: based on position and recognition-based markers or objects.

## Design

In order to decide what kind of systems we must use, we rely on what characteristics should this project collect:

- The system should be intuitive, using terminal's camera.
- Not necessarily being close to the object.
- The item you want to act with doesn't need to take nothing added or, if needed, be at less shocking as possible.
- The complexity of its installation should be as low as possible.

Be based on these characteristics described the best option is the use of markers / stickers with view to that in the future we be able to recognize objects directly.

Once we know how to identify the object, is needed to show the user a kind of selector, maybe a box or button around the object overlapping the image, that is, use the Augmented Reality.

After selecting the object you want to act with, a petition to a server should be made by a client.

As a proof of concept, the server store the status information and also is acts on the objects that are available in this work through a graphic interface, a simulator which displays a virtual world that you can control.

It should be emphasized that the philosophy of this project is based on the principle of Open Source. This expression is how it is known the software that is freely developed and distributed, focusing in practical benefits.

For this project, the entire system has been developed under the license of Google, using both Android and AndAR libraries and also GSON's, even Apache Tomcat licence.



Figura 5: Final application view.

## Conclusions

If the author of this paper could have had sufficient time to expand, broaden and add more options, he would have achieved the ideal solution, which was supposed as the initial seed of the project.

The immediate future consists of distinguishing objects by their physical attributes with the help of the camera installed in the user mobile device.

Our approach consisted in creating the recognition system of markers.

Never the less the author would wish to offer the sequence of procedures and applications aimed to develop a more attractive interface:

- A Adding a progress bar up and down to determine the strength lights.
- If the elements require a position indicator, it is would be necessary to offer an animate selecting system (as for shutters, doors or windows).
- With the elements that can be recognized, being impossible to show their principal interface (like a computer or a television set), it would be necessary to add the viewing in real time and to duplicate the images generated in the computer on the device display.
- Add radio or music player controller, with buttons like “play”, “pause” or others.
- Print using network printers.
- Pets recognition, asking for information about their state, or quoting with their veterinarian.
- Etc.

Moreover, the increasing number of smart houses provides the possibility to develop some options to be added, aimed at the recognition and the identification of the proprieties of the object preconfigured by its manufacturer, like doing Internet shopping from the refrigerator or turning on the household alarm.



# Índice de figuras

|   |    |
|---|----|
| 2.1. Arquitectura Centralizada . . . . .  | 10 |
| 2.2. Arquitectura Descentralizada . . . . .   | 10 |
| 2.3. Arquitectura Distribuida . . . . .   | 10 |
| 2.4. Arquitectura Híbrida . . . . .   | 10 |
| 2.5. Representación de un 1 lógico en el protocolo X10 . . . . .                                | 12 |
| 2.6. Códigos paquete X10 . . . . .  | 13 |
| 2.7. Ejemplos de dispositivos X10: DIN, Enchufe, Empotrable, Mi-<br>cromódulo . . . . .         | 14 |
| 2.8. Diferencia entre las conexiones X10 y EIB . . . . .  | 15 |
| 2.9. Algunos ejemplos de elementos EIB . . . . .  | 17 |
| 2.10. Sistema EIB total . . . . .   | 18 |
| 2.11. Sistemas domóticos instalados en 2007 en viviendas de nueva<br>promoción . . . . .        | 19 |
| 2.12. Progresión de la cifra de mercado de viviendas inteligentes<br>entre 2009 y 2012. . . . . | 19 |
| 2.13. AR vs VR . . . . .  | 21 |
| 2.14. Vista desde un terminal con Wikitude, Layar y Junaio . . . . .                            | 23 |
| 2.15. Marcador base de AR . . . . .   | 23 |
| 2.16. Ejemplos de marcadores de AR . . . . .  | 24 |
| 2.17. Ejemplo de RA sobre marcador . . . . .  | 25 |
| 3.1. Diagrama del sistema. . . . .  | 32 |
| 3.2. Ejemplo tabla MySQL . . . . .  | 39 |
| 4.1. Arquitectura del proyecto. . . . .   | 44 |
| 4.2. Funcionamiento del servidor Apache Tomcat. . . . .   | 45 |
| 4.3. Contenido de la carpeta “webapps”. . . . .   | 45 |
| 4.4. Contenido del archivo “lampara”. . . . .   | 45 |
| 4.5. Ciclo de vida de un objeto Activity de Android. . . . .                                    | 48 |
| 4.6. Vista del interfaz de usuario sin ningún marcador. . . . .                                 | 49 |
| 4.7. Vista del interfaz de usuario con marcadores. . . . .                                      | 50 |

|   |    |
|---|----|
| 4.8. Vista del interfaz de usuario con lista de opciones. . . . .   | 50 |
| 4.9. Diagrama de clases java del servidor. . . . .                  | 52 |
| 4.10. Vista del Mundo Virtual. . . . .                              | 53 |
| 4.11. Vista del menú de control de la base de datos. . . . .        | 54 |
| 4.12. Vista de las tablas que existen en la base de datos. . . . .  | 57 |
| 4.13. Descripción de la tabla IDandOBJECTS. . . . .                 | 57 |
| 4.14. Vista del contenido de la tabla IDandOBJECTS. . . . .         | 57 |
| 4.15. Vista de la descripción de la tabla “lampara”. . . . .        | 58 |
| 4.16. Vista del contenido de la tabla “lampara”. . . . .            | 58 |
| 6.1. Objetos vistos con la solución de Google al reto ImageNet. . . | 74 |
| 6.2. Ejemplo de uso futuro de R.A. . . . .                          | 75 |
| 6.3. Casa inteligente . . . . .                                     | 76 |
| 6.4. Habitación vista desde el terminal en un futuro. . . . .       | 80 |

# Índice de cuadros

|   |    |
|---|----|
| 4.1. Escenarios de pruebas. . . . .                 | 59 |
| 5.1. Fases del proyecto y horas invertidas. . . . . | 65 |
| 5.2. Presupuesto de SoftWare y HardWare. . . . .    | 66 |
| 5.3. Presupuesto total. . . . .                     | 66 |
| 5.4. Planificación de tareas. . . . .               | 69 |





# Índice general

|   |           |
|---|-----------|
| <b>1. Introducción</b>                                  | <b>1</b>  |
| 1.1. Motivación del proyecto . . . . .                  | 1         |
| 1.2. Planteamiento del problema . . . . .               | 2         |
| 1.3. Objetivos . . . . .                                | 4         |
| 1.4. Fases de desarrollo . . . . .                      | 5         |
| 1.5. Contenido de la memoria . . . . .                  | 6         |
| <b>2. Estado del arte</b>                               | <b>7</b>  |
| 2.1. Domótica . . . . .                                 | 7         |
| 2.1.1. Prestaciones y Aplicaciones . . . . .            | 8         |
| 2.1.2. Protocolos . . . . .                             | 11        |
| 2.1.3. Actualidad . . . . .                             | 18        |
| 2.2. Realidad aumentada . . . . .                       | 20        |
| 2.2.1. Realidad Aumentada VS Realidad Virtual . . . . . | 20        |
| 2.2.2. Tipos de RA . . . . .                            | 21        |
| 2.2.3. Aplicaciones . . . . .                           | 25        |
| 2.2.4. Actualidad . . . . .                             | 28        |
| <b>3. Diseño</b>  | <b>31</b> |
| 3.1. Punto de partida. . . . .                          | 31        |
| 3.2. Reconocimiento de objetos . . . . .                | 33        |
| 3.3. Realidad Aumentada . . . . .                       | 35        |
| 3.4. Cliente . . . . .                                  | 36        |
| 3.5. Servidor . . . . .                                 | 36        |
| 3.5.1. Estructura de ficheros . . . . .                 | 37        |
| 3.6. Base de datos . . . . .                            | 38        |
| 3.7. Simulador: un mundo virtual . . . . .              | 39        |
| 3.8. Restricciones y marco regulador . . . . .          | 39        |

|   |           |
|---|-----------|
| <b>4. Trabajo realizado: arquitectura del prototipo</b> | <b>43</b> |
| 4.1. Arquitectura . . . . .                             | 43        |
| 4.2. Descripción funcional . . . . .                    | 46        |
| 4.2.1. Cliente . . . . .                                | 46        |
| 4.2.2. Servidor . . . . .                               | 46        |
| 4.3. Descripción operacional . . . . .                  | 47        |
| 4.3.1. Cliente . . . . .                                | 47        |
| 4.3.2. Servidor . . . . .                               | 51        |
| 4.4. Implementación . . . . .                           | 54        |
| 4.4.1. Cliente . . . . .                                | 54        |
| 4.4.2. Servidor . . . . .                               | 55        |
| 4.5. Escenarios de pruebas . . . . .                    | 58        |
| <b>5. Historia del proyecto</b>                         | <b>61</b> |
| 5.1. Fases de desarrollo . . . . .                      | 61        |
| 5.2. Presupuesto . . . . .                              | 63        |
| 5.3. Planificación . . . . .                            | 67        |
| 5.4. Reflexiones personales . . . . .                   | 67        |
| 5.5. Diagrama de Gantt . . . . .                        | 68        |
| <b>6. Conclusiones y trabajo futuro</b>                 | <b>71</b> |
| 6.1. Consecución de objetivos . . . . .                 | 71        |
| 6.2. Reflexión personal sobre el futuro . . . . .       | 73        |
| 6.2.1. Reconocimiento de objetos . . . . .              | 73        |
| 6.2.2. Realidad Aumentada . . . . .                     | 74        |
| 6.2.3. Casas inteligentes . . . . .                     | 75        |
| 6.2.4. Internet de la cosas . . . . .                   | 76        |
| 6.3. Líneas de trabajo futuro . . . . .                 | 78        |
| 6.3.1. Reconocimiento de objetos . . . . .              | 78        |
| 6.3.2. Realidad Aumentada . . . . .                     | 78        |
| 6.3.3. Domótica: casas inteligentes . . . . .           | 79        |
| 6.3.4. Internet de las cosas . . . . .                  | 79        |
| <b>7. Apéndices</b>                                     | <b>81</b> |
| 7.1. Manual de instalación . . . . .                    | 81        |
| 7.2. Manual de usuario . . . . .                        | 82        |

# Capítulo 1

## Introducción

En los siguientes apartados se describe la motivación que ha dado forma a este Trabajo de Fin de Grado, su planteamiento y sus objetivos, así como una breve descripción de su contenido.

### 1.1. Motivación del proyecto

La llegada de los smartphones, y dispositivos móviles en general, al mercado ha dado una nueva acepción al término “estar conectados”. Ya no sólo podemos contactar con otras personas, sino que además queremos poder hacerlo las 24 horas del día: queremos estar continuamente conectados.

Nuestro entorno está conectado a Internet, podemos consultar casi cualquier duda a través de nuestros dispositivos de casi cualquier elemento de nuestro alrededor, comprar entradas, hacer la compra o consultar el tiempo: podemos interactuar con el entorno usando nuestros terminales. ¿Por qué no hacer esto en nuestros hogares?

Imaginemos que podemos conectar cada elemento de nuestra casa a una pequeña red a la que siempre estemos conectados. Podríamos encender cualquier aparato, subir o bajar las persianas, dar de comer a nuestras mascotas... Pero, ¿y si además de hacerlo posible, lo hacemos sencillo?

Las cámaras integradas en los terminales son una realidad que todo usuario exige como mínimo, ¿por qué no usarla para hacer esta interacción más intuitiva y eficiente?

Interactuar con nuestros hogares lleva siendo una realidad desde hace más de 20 años. La televisión a la carta, el control de acceso a nuestro edificio

o subir y bajar persianas o toldos es un hecho que ya ha pasado a ser algo cotidiano en nuestro día a día.

Sin embargo, ha ido creciendo la necesidad de interactuar con cada vez más objetos. Poder escuchar música en cada habitación, controlar las luces o encender la calefacción antes de llegar a casa, a través de un simple SMS<sup>1</sup>, son pequeños ejemplos de lo que ya es habitual.

En un intento de ir más allá, se propone en este documento una solución con la que se puede controlar hasta el más mínimo elemento que se desee, a través de una interfaz clara y sencilla.

Para ello, la Realidad Aumentada hace alarde de su desarrollo, reconociendo el entorno y planteando esta posibilidad ante nuestros ojos, usando la cámara de nuestro dispositivo para controlar nuestra casa, incluyendo cualquier aparato electrónico u otros elementos no electrónicos como puertas, ventanas, toldos o persianas.

## 1.2. Planteamiento del problema

El escenario en el que nos encontramos es el siguiente: poder reconocer nuestro alrededor en el hogar para poder interactuar con cada elemento, es decir, saber qué hay y qué podemos hacer con ello, así como llevar a cabo esa acción a través de una presentación al usuario atractiva, clara e intuitiva como es el uso directo de la imagen que se ve a través de la cámara de nuestro dispositivo móvil, es decir, poder observar el mundo real a través de nuestra cámara y poder seleccionar el objeto u objetos sobre los que queremos actuar para cambiar su estado.

Para ello, es necesario saber qué hay a nuestro alrededor, qué podemos hacer con esos objetos, y por supuesto, llevar a cabo las diferentes acciones. Es decir, hace falta:

- Desarrollar un sistema de reconocimiento de objetos. ¿Qué hay a nuestro alrededor?
- Desarrollar un sistema de control sobre las acciones de los distintos objetos. ¿Qué podemos hacer con esos elementos?

---

<sup>1</sup>Short Message Service

- Desarrollar un sistema de actuación sobre éstos. ¿Cómo realizamos la acción?

Lo primero que se debe decidir es la forma de reconocer nuestro alrededor y diferenciar los elementos que se encuentren. Para encontrar cada elemento existen diferentes opciones:

- Que cada uno de ellos emita señales (como el uso de iBeacons[1]) y que nuestro dispositivo sepa interpretarlas, ya sean radio o por cable a un servidor que más tarde envía esa información al terminal.
- Conociendo su posición a través de un modelado del entorno con un sistema de coordenadas relativo.
- Interactuación de forma cercana: NFC[2] o RFID[3].
- Uso de Infrarojos.
- Reconocimiento visual a través de etiquetas o marcadores o del propio objeto.

También es necesario saber qué podemos hacer con cada uno de estos elementos que se han descubierto. Para ello caben diferentes posibilidades.

- Que el propio objeto guarde esta información y la remita al terminal cuando se solicite.
- Que el objeto en cuestión tenga asignado un código identificador, el cual, una vez reconocido, implique una serie de acciones distintas de las de otros elementos. No obstante, esto a su vez conlleva varias opciones:
  - Que el propio dispositivo móvil guarde esta información.
  - Que un servidor, que hace las veces de controlador, guarde esta información y que se pueda consultar.

Por último, una vez sabido qué hay y qué hacer, llevar a cabo la acción a través de los actuadores pertinentes.

A todo esto se suma el hecho de que el usuario no tiene por qué conocer toda la tecnología que hay detrás de este planteamiento, luego, el manejo de este sistema debe ser sencillo y nuevamente, intuitivo. Hace falta desarrollar un interfaz de control del sistema claro. Existen diferentes opciones a la hora de elegir ese interfaz, como son:

- Presentación a través de listados desplegados o no desplegados.
- Presentación a través de navegación por categorías (luces, electrodomésticos, etc).
- Presentación en imágenes estáticas y/o dinámicas.
- Presentación a través del uso de Realidad Aumentada a través de la cámara de un dispositivo.

### 1.3. Objetivos

Se quiere llevar a cabo una aplicación sobre sistema Android[4], a modo de prueba de concepto, que sea de fácil manejo para el control domótico del hogar.

Para llegar a ello se pueden dividir los objetivos en dos tipos: generales y específicos.

- Objetivos generales:
  1. Implementar un sistema cómodo, ya que el aparato de control es un terminal y no requiere más infraestructura, para interactuar con el entorno del hogar.
  2. Desarrollar un sistema de control sobre los objetos que sea ampliable de forma sencilla y fácil de implementar.
  3. Desarrollar un sistema de identificación y reconocimiento de objetos simple.
  4. Desarrollar un interfaz de usuario intuitivo y de fácil manejo para el control del sistema.
- Objetivos específicos:
  1. Conocer el sistema Android y el uso de la biblioteca AndAR[5].
  2. Integrar funcionalidades de Realidad Aumentada proporcionadas por Google en una aplicación Android.
  3. Desarrollar una base de datos de fácil manejo y uso, así como un sistema de ficheros donde guardar la información de estado del entorno del hogar.

## 1.4. Fases de desarrollo

Este Trabajo de Fin de Grado se ha dividido en las siguientes fases de desarrollo:

1. Análisis del problema: concreción del escenario y los objetivos a cumplir.
2. Análisis de las soluciones: concreción de las partes necesarias para resolver el problema.
3. Diseño de las diferentes partes: control domótico y realidad aumentada.
4. Diseño de una base de datos para el control domótico: elección de la información a guardar. Desarrollo de un programa de uso sencillo para el manejo de la base de datos.
5. Servidor: ocupado en resolver las consultas del cliente, así como manejar la información de los diferentes ficheros.
6. Cliente: realiza las consultas e interpreta las respuestas del servidor.
7. Interfaz del servidor: muestra los cambios que se realizan en el entorno a través de imágenes que cambian según el estado de los objetos.
8. Interfaz del cliente: muestra los objetos con los que se puede interactuar, así como sus diferentes acciones posibles.
9. Pruebas Cliente - Servidor: respecto a la petición y resolución de respuestas.
10. Desarrollo de un motor de reconocimiento de etiquetas (relativas a sendos objetos).
11. Pruebas de reconocimiento: a través de una aplicación Android leer e interpretar las diferentes etiquetas.
12. Interfaz final: aplicación final, de uso sencillo e intuitivo, que auna las dos partes anteriormente diferenciadas.
13. Pruebas finales: realizar diferentes pruebas para depuración del programa, con diferentes individuos no necesariamente familiarizados con el tema.



## 1.5. Contenido de la memoria

### **Primera parte:**

Introducción: En esta parte se explica la motivación que ha hecho llevar a cabo este trabajo, un análisis del escenario propuesto concluyendo en problemas y objetivos que solventar, además de una explicación sobre las fases del trabajo realizado para dar solución a estos problemas, así como un resumen sobre el contenido de esta memoria.

### **Segunda parte:**

Estado del arte: Se hace un repaso de la actualidad en cuanto a la presencia de la domótica y el uso de la Realidad Aumentada en la vida diaria.

### **Tercera parte:**

Diseño: En esta parte se hace un análisis sobre el diseño adoptado, concluyendo en el porqué de esta solución a través de comparaciones con otras tecnologías y opciones.

### **Cuarta parte:**

Arquitectura del prototipo: En este apartado se repasan las diferentes partes del trabajo, describiendo cada una de forma extensa y definiendo las diferentes estructuras que se han usado hasta llegar al resultado final.

También se repasan las diferentes pruebas realizadas a lo largo del desarrollo del trabajo.

### **Quinta parte:**

Historia del proyecto: Un repaso sobre la cronología que ha llevado este trabajo.

### **Sexta parte:**

Conclusiones: Se hace una evaluación del proyecto repasando sus objetivos y su cumplimiento.

El futuro: En esta parte se da una vista hacia el futuro planteando las diferentes posibilidades de ampliación que tiene este trabajo.

### **Séptima parte:**

Apéndices: Por último, se recoge una pequeña documentación sobre la instalación y configuración de las diferentes partes, así como de su uso por parte del usuario.

# Capítulo 2

## Estado del arte

En los siguientes apartados de este capítulo veremos el estado actual de este tipo de tecnologías, empezando por el control domótico y continuando con la Realidad Aumentada.

### 2.1. Domótica

Domótica: del latín *domus*, casa e *informática*

La Asociación Española de Domótica e Inmótica [6] define la domótica como “(...) el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema.”

La historia de la domótica empieza en los años 70 cuando se comenzó a automatizar edificios, en cuanto al control sobre actuadores. Pero no fue realmente hasta los 80 cuando empieza a parecerse al concepto que tenemos ahora, de un uso en el hogar sobre elementos más pequeños y sencillos. Según avanzaron los años y aparecieron diferentes tecnologías se empezó a diseñar edificios, tanto industriales como de oficinas, así como de viviendas, teniendo en cuenta las posibilidades que aportaba la domótica.

Con el auge de la informática, se permite una expansión de este tipo de sistemas, permitiendo incorporar, a los hogares y edificios, cableado para las conexiones.

El primer programa desarrollado, entorno a la domótica, fue el conocido como S.A.V.E.[7], donde se destaca la domótica y los sistemas de control de

los edificios para obtener una mayor eficiencia, tanto en las operaciones de control como en el consumo de energía.

El termino se acuñó en Francia por primera vez en 1988, cuando los diccionarios franceses aceptaban el término *domotique* como “*el concepto de vivienda que integra todos los automatismos en materia de seguridad, gestión de la energía, comunicaciones y otros servicios*”

Inicialmente, la única instalación domótica que se podía hacer, consistía en el uso de sensores y actuadores que se unían con una tecnología centralizada a un autómatas. Se trataba de una arquitectura poco flexible, que resultaba en que las posibles ampliaciones fuesen costosas y difíciles.

Sin embargo, según nos acercamos a mediados de los años 90, el precio del Hardware descende, con lo que se pueden multiplicar los puntos de control, así como mejorar la tecnología, pasando a sistemas distribuidos que dividen el trabajo. Estos, están entre sí conectados a través del protocolo X-10, un protocolo desarrollado por Pico Electronics[8] en Glenrothes, Escocia.

No obstante, no es el único protocolo que se ha usado, o se usa en el presente, para hacer posible el intercambio de información entre los diferentes componentes. Entre los más usados encontramos el protocolo KNX/EIB en el marco europeo y Lonworks en Estados Unidos.

La mayor diferencia entre estos protocolos es que el X-10 (véase apartado 2.1.2) lo soporta equipos baratos y de fácil instalación, sin embargo, en instalaciones complejas el precio aumenta y el rendimiento se degrada ya que las opciones que ofrece son limitadas. Por otra parte, el protocolo KNX/EIB sí puede realizar instalaciones más complejas sin aumentar excesivamente el precio de partida, pero éste ya es suficientemente alto, ya que requiere un cableado dedicado (véase el apartado 2.1.2 ).

### 2.1.1. Prestaciones y Aplicaciones

#### Prestaciones

Cuando se habla de domótica, automáticamente nos viene a la mente “control de nuestro hogar”. En resumen, en eso consiste la domótica. Ahora bien, ¿qué opciones reales nos da esta tecnología?

El objetivo mayor de la domótica es el de provocar un aumento de la calidad de vida, asegurando una serie de particularidades que sean controlables de forma automática. De alguna manera, asegurar una vida más sencilla al cliente, al mismo tiempo que éste sienta un confort, una seguridad, un control

de su hogar y sea capaz de comunicarse con él, así como un aprovechamiento de la energía.

La facilidad de uso, la ampliabilidad, la sencillez de instalación y su flexibilidad son las características que un sistema domótico debe aunar, desembocando en un ahorro de energía y, sobre todo, económico.

Por otra parte, los usuarios y técnicos deben evaluar esta tecnología para que evolucione y crezca. A la par que se puede convertir en un servicio flexible y adaptable a cada usuario y, al mismo tiempo, sea sencillo hacerlo por parte de los técnicos.

El usuario querrá una simplicidad en su uso, que haya variedad en elementos controlables y sus funcionalidades. Además, un buen servicio postventa es básico para poder resolver cualquier situación de duda o contratiempo. La posible ampliación del sistema también se debe contemplar para que el usuario pueda ir añadiendo módulos a su sistema en el hogar.

Los técnicos tienen en cuenta la topología del conjunto, tipo de arquitectura, velocidad y medios de transmisión así como el protocolo que se puede usar, ya que todos estos tecnicismos deben dar una solución elegante y sencilla.

Existen cuatro tipos principales de arquitecturas de control domótico:

1. Centralizada: Un único controlador recibe toda la información necesaria para actuar sobre los elementos, enviando la información a los actuadores e interfaces.
2. Descentralizada: Existen varios controladores que intercambian información entre ellos a través de un bus, dividiendo el trabajo por zonas de control.
3. Distribuida: Todos los elementos están conectados a un bus, de manera que cada uno de ellos tiene su propio controlador, actuador e interfaz, e intercambia datos con los demás elementos a través de dicho bus.
4. Híbrida: Mezcla las tres arquitecturas anteriores dependiendo de la complejidad de cada elemento o conjunto de elementos.

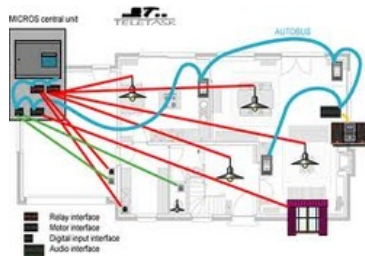


Figura 2.1: Arquitectura Centralizada

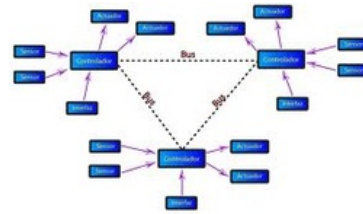


Figura 2.2: Arquitectura Descentralizada



Figura 2.3: Arquitectura Distribuida



Figura 2.4: Arquitectura Híbrida

Hacer o ampliar su instalación supone diferentes problemas para cada tecnología. Mientras que para la centralizada y descentralizada supone una configuración de los controladores que congregan varios actuadores, con su correspondiente poca flexibilidad, ya que se debe tener en cuenta una información de forma manual, para la distribuida cada elemento hace el trabajo de estos controladores, intercambiando la información necesaria para su configuración, lo que se denomina “Plug and Play” [9].

## Aplicaciones

Las opciones de control y las aplicaciones de la domótica van desde lo más sencillo, como interactuar con elementos del entorno con comandos ya predeterminados (como la apertura o cierre de una puerta, o la programación de un programa de algún electrodoméstico), hasta añadir opciones variadas como poder alimentar a nuestras mascotas o cambiar el hilo musical, pasando por seguridad, comunicación o sistema de riego.

Algunas de estas aplicaciones son:

- Control del suministro de agua, gas o electricidad.
- Sistemas de seguridad.
- Sistema de control de la música ambiente.
- Sistema de riego automático.
- Control de climatización e iluminación.
- Cerradura electrónica.
- Circuito cerrado interior de televisión integrado en la TV convencional.
- Electrodomésticos programables y conectables a un bus de datos.
- Puertas automáticas, que se abren por presencia. Control de subida y bajada de persianas y apertura de ventanas.
- Suelo radiante.
- Detectores de movimiento.
- Etc.

### 2.1.2. Protocolos

#### X-10

El protocolo X-10 es un estándar para poder establecer la comunicación entre diferentes componentes compatibles con los que se controlan dispositivos como luces o electrodomésticos.

Se trata del protocolo más difundido en el mundo de la domótica ya que fue el primero. Desarrollado entre los años 74 y 75 del siglo XX, su éxito se basa en su objetivo principal: transmitir datos por líneas de baja tensión a una baja velocidad, de entre 50 y 60 bits por segundo (insuficiente para el intercambio de información de hoy en día si el sistema es grande), con lo que no hace falta añadir cableado al del hogar, ya que puede usar sus líneas de fuerza.

Los diferentes fabricantes pueden hacer uso de este protocolo, pero deben basarse en el trabajo desarrollado por Pico Electronics, es decir, deben incluir sus elementos ya que tienen la patente, resultando en un precio muy

competitivo y que, tras 30 años de uso, las pequeñas empresas o particulares pueden hacer sus instalaciones sin prácticamente conocimientos previos. Sin ir más lejos, los productos antiguos siguen funcionando a día de hoy ya que se busca una filosofía de interacción y compatibilidad hacia atrás, con lo que los quebraderos de cabeza, en este sentido, no existen.

**Funcionamiento** Las señales de control se basan en ráfagas de pulsos que representan 1's y 0's. Por otro lado, estas señales se transmiten en la red local, de 50Hz. Además la representación de un 1 será la presencia de un pulso de 120Hz en el paso por 0 del primer semiciclo, repitiéndose dos veces más, a modo de redundancia, en cada tercio de este semiciclo positivo. Además, coincide con el paso por cero de las tres fases de un sistema trifásico como es el del hogar (Véase la figura 2.5).

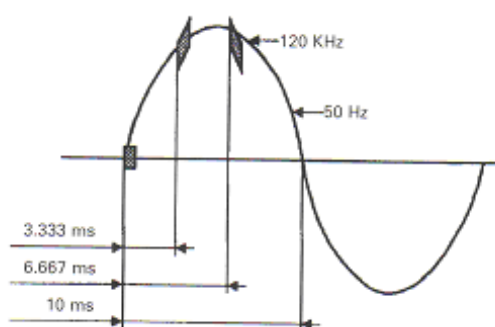


Figura 2.5: Representación de un 1 lógico en el protocolo X10

Un código X10 implica 11 ciclos de red divididos en:

1. **[Ciclos 0 y 1]:** Código de Inicio: Cabecera de inicio de paquete.
2. **[Ciclos 2 al 5]:** Código de casa: Forma parte de la redirección del paquete.
3. **[Ciclos 6 al 10]:** Código de función o el código numérico: Con el código de casa redirecciona el paquete al módulo final.

La figura 2.6 incluye los diferentes códigos.

|                              | Códigos de Casa |    |    |    |    | Códigos de Unidad |    |    |    |     |
|------------------------------|-----------------|----|----|----|----|-------------------|----|----|----|-----|
|                              | H1              | H2 | H4 | H8 |    | D1                | D2 | D4 | D8 | D16 |
| A                            | 0               | 1  | 1  | 0  | 1  | 0                 | 1  | 1  | 0  | 0   |
| B                            | 1               | 1  | 1  | 0  | 2  | 1                 | 1  | 1  | 0  | 0   |
| C                            | 0               | 0  | 1  | 0  | 3  | 0                 | 0  | 1  | 0  | 0   |
| D                            | 1               | 0  | 1  | 0  | 4  | 1                 | 0  | 1  | 0  | 0   |
| E                            | 0               | 0  | 0  | 1  | 5  | 0                 | 0  | 0  | 1  | 0   |
| F                            | 1               | 0  | 0  | 1  | 6  | 1                 | 0  | 0  | 1  | 0   |
| G                            | 0               | 1  | 0  | 1  | 7  | 0                 | 1  | 0  | 1  | 0   |
| H                            | 1               | 1  | 0  | 1  | 8  | 1                 | 1  | 0  | 1  | 0   |
| I                            | 0               | 1  | 1  | 1  | 9  | 0                 | 1  | 1  | 1  | 0   |
| J                            | 1               | 1  | 1  | 1  | 10 | 1                 | 1  | 1  | 1  | 0   |
| K                            | 0               | 0  | 1  | 1  | 11 | 0                 | 0  | 1  | 1  | 0   |
| L                            | 1               | 0  | 1  | 1  | 12 | 1                 | 0  | 1  | 1  | 0   |
| M                            | 0               | 0  | 0  | 0  | 13 | 0                 | 0  | 0  | 0  | 0   |
| N                            | 1               | 0  | 0  | 0  | 14 | 1                 | 0  | 0  | 0  | 0   |
| O                            | 0               | 1  | 0  | 0  | 15 | 0                 | 1  | 0  | 0  | 0   |
| P                            | 1               | 1  | 0  | 0  | 16 | 1                 | 1  | 0  | 0  | 0   |
| Apaga Todas las Unidades     |                 |    |    |    |    | 0                 | 0  | 0  | 0  | 1   |
| Encender Todas las Luces     |                 |    |    |    |    | 0                 | 0  | 0  | 1  | 1   |
| Encender                     |                 |    |    |    |    | 0                 | 0  | 1  | 0  | 1   |
| Apagar                       |                 |    |    |    |    | 0                 | 0  | 1  | 1  | 1   |
| Atenuar Intensidad           |                 |    |    |    |    | 0                 | 1  | 0  | 0  | 1   |
| Aumentar Intensidad          |                 |    |    |    |    | 0                 | 1  | 0  | 1  | 1   |
| Apagar Todas las Luces       |                 |    |    |    |    | 0                 | 1  | 1  | 0  | 1   |
| Código Extendido             |                 |    |    |    |    | 0                 | 1  | 1  | 1  | 1   |
| Petición de Saludo           |                 |    |    |    |    | 1                 | 0  | 0  | 0  | 1   |
| Aceptación de Saludo         |                 |    |    |    |    | 1                 | 0  | 0  | 1  | 1   |
| Atenuación Preestablecida    |                 |    |    |    |    | 1                 | 0  | 1  | x  | 1   |
| Datos Extendidos (analógico) |                 |    |    |    |    | 1                 | 1  | 0  | 0  | 1   |
| Estado = on                  |                 |    |    |    |    | 1                 | 1  | 0  | 1  | 1   |
| Estado = off                 |                 |    |    |    |    | 1                 | 1  | 1  | 0  | 1   |
| Petición de Estado           |                 |    |    |    |    | 1                 | 1  | 1  | 1  | 1   |

Figura 2.6: Códigos paquete X10

Existen tres tipos de dispositivo X10: transmisores, receptores y bidireccionales (además de un cuarto inalámbrico).

Los transmisores pueden direccionar hasta 256 receptores. Los receptores se pueden configurar para que varios respondan a la misma señal. Los dispositivos bidireccionales pueden responder a las peticiones de los transmisores, pudiendo también informar del correcto funcionamiento de la orden dada.

Algunos ejemplos de dispositivos X10 los podemos encontrar en la figura 2.7





Figura 2.7: Ejemplos de dispositivos X10: DIN, Enchufe, Empotrable, Micromódulo

## KNX/EIB

El Bus de Instalación Europeo, o EIB, es el siguiente paso desde el punto de vista de la escalabilidad, ya que está destinado a un uso en el total de un edificio, así como para casas unifamiliares o conjunto de viviendas.

EIB surge con la idea de introducir en el mercado un sistema unificado para la gestión de edificios. Creado por la asociación AEIB, con sede en Bruselas, quien emite las marcas registradas con el sistema, así como estándares de dispositivos y calidad.

A diferencia de X10, EIB necesita un cableado añadido, que en la planificación de edificaciones se puede incluir, mientras que las construcciones ya finalizadas su instalación implicaría una reestructuración, una obra y un coste derivado. Este cableado extra es, sin ir más lejos, un Bus de control paralelo al cableado del edificio. A este bus están conectadas todas las partes, lo que deriva en una falta de necesidad de una unidad central de control, como un ordenador o servidor. Existen tres posibles medios físicos para interconectar dispositivos: el cable par trenzado, red eléctrica de baja tensión y en un futuro próximo dispositivos de radio frecuencia.

En sí, este cableado presenta un ahorro en cantidad de despliegue, reduciendo el cableado y el coste asociado, ya que realmente es más sencillo conectar todos los elementos a una única autopista de datos, mientras que en X10 es necesario conectar entre sí los elementos que necesiten hablar “face to face”.

La figura 2.8 muestra una comparación en el número de conexiones entre X10 y EIB

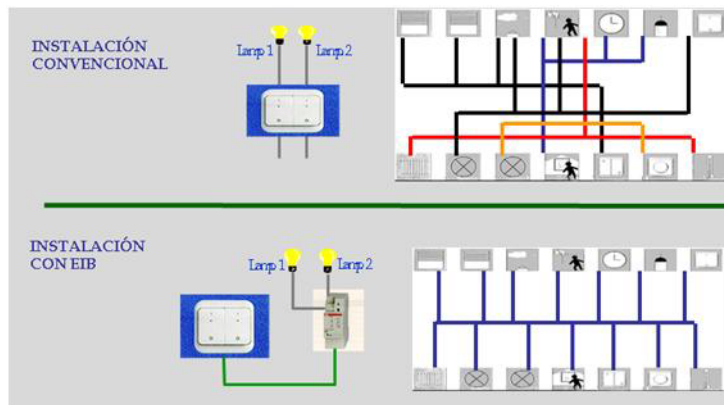


Figura 2.8: Diferencia entre las conexiones X10 y EIB

Para configurar este sistema, basta con un ordenador que se conecte al bus y programe las diferentes partes. Una vez terminado este trabajo, se puede dejar al servicio funcionar de forma automática.

La arquitectura EIB está basada en el modelo de referencia OSI[10], haciendo uso de los niveles Físico, Enlace, Red y Aplicación directamente.

- Nivel de aplicación: Provee funcionalidades que pueden ser usadas directamente por los programas de aplicación. Cabe distinguir que cada elemento EIB tiene un nivel de aplicación diferente dependiendo de sus necesidades, sea receptor, emisor o ambas.
- Nivel de red: En el caso de EIB el bus se divide en áreas y líneas indicadas en direcciones físicas. Filtra los telegramas mirando la dirección destino, permitiendo su paso o no.
- Nivel de enlace: Los mensajes necesitan de un control de acceso al medio, notificación de errores y control de flujo, así como de un direccionamiento físico.
- Nivel físico: Interpreta y convierte en señales física como voltajes y corrientes, ondas electromagnéticas para finalmente transmitirlos por el medio.

No necesita los demás niveles ya que no está orientado a conexión. No necesita asegurarse de la llegada de datos, por lo que no requiere nivel de transporte. Sus mensajes son cortos, así que no necesita un sistema de control de sesión. Además, no hay forma de presentación de datos, luego no requiere un nivel de presentación.

El siguiente paso en la evolución de este sistema es KNX, que usa la pila de comunicación de EIB, ampliando capas físicas, configuración y aplicación.

Existen diferentes medios de comunicación física:

- Cableado de par trenzado: EIB.TP, a 9600bps, usa CSMA/CA[11] para evitar colisiones y reintentos.
- Red eléctrica: EIB.PL, con corrientes portadoras de 230V a 1200 / 2400 bps. Usa SFSK[12] como modulación.
- Ethernet: EIB.net, sirve de Backbone entre segmentos EIB, además de permitir el uso del protocolo IP[13] para transferencia de paquetes.
- Radio: EIB.RF, usando portadoras y repetidores.
- Infrarojo: EIB.IR, para el uso de mandos a distancia.

El hardware necesario se puede dividir en 4 tipos de elementos:

- Actuadores: actúan sobre los elementos, el último paso justo antes del cambio en el estado del elemento controlado.
- Sensores: Recogen datos desde pulsadores hasta anemómetros, monitorizando el sistema a través de pantallas, muchas de ellas táctiles.
- Pasarelas: Permiten conectarse en remoto con el sistema o con otros elementos.
- Acopladores: Introducen la señal y realizan la separación física.

Como las instalaciones que rodean al protocolo EIB son de un tamaño considerable, sus equipos también lo son. En la figura 2.9 algunos ejemplos de elementos EIB.

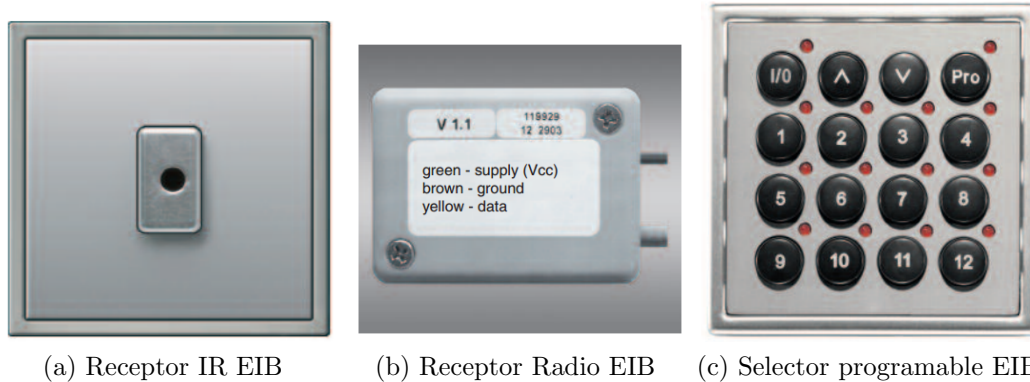


Figura 2.9: Algunos ejemplos de elementos EIB

**Topología** Para interconectar los diferentes dispositivos existen dos tipos de topologías:

- **Línea:** Se pueden conectar hasta 64 dispositivos (en función de la fuente de alimentación).
- **Área:** Se pueden conectar hasta 15 líneas, haciendo un total de 960 dispositivos.
- **Sistema completo:** A través de una línea principal se pueden unir hasta 15 áreas, llegando a acumular hasta 14.400 elementos.

La figura 2.10 muestra el sistema completo EIB.

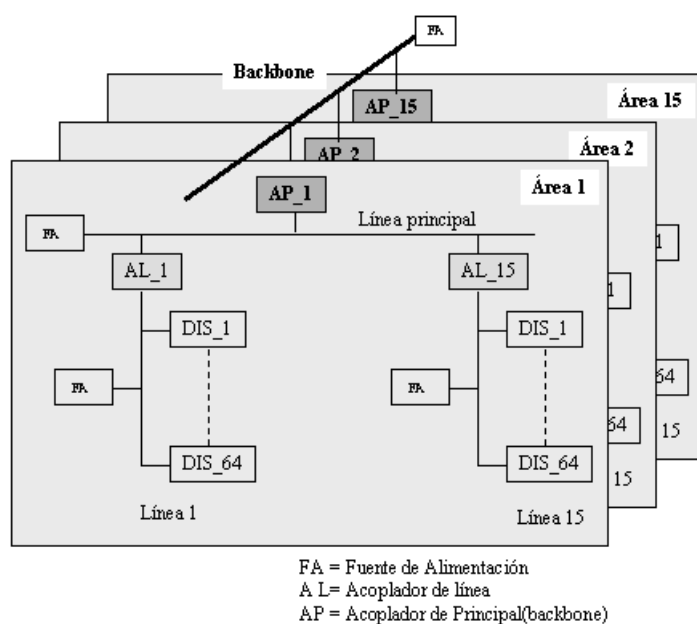


Figura 2.10: Sistema EIB total

### 2.1.3. Actualidad

En España, la Asociación Española de Domótica e Inmótica, se encarga de aumentar la implantación de la domótica e inmótica a través de la promoción de la tecnología.

CEDOM nace en 1992 como el Comité Español de la Domótica. En el año 2001 se abre no sólo a fabricantes sino también al resto de colectivos, instaladores y desarrolladores, cambiando su nombre pero no así sus siglas.

Desde entonces se ha ido adaptado a los cambios y dificultades que ha sufrido el sector, siendo en la actualidad la única asociación a nivel nacional que reúne a todos los agentes <sup>1</sup> del sector: fabricantes, equipos auxiliares, distribuidores, integradores, centros tecnológicos y de formación, desarrolladores, universidades y medios de comunicación.

En 2007, el número de sistemas instalados en viviendas de nueva promoción fue de más de cuarenta y siete mil, que sumaron más de 120 millones de euros. Ver figura 2.11.

<sup>1</sup>Lista de socios de CEDOM en [www.cedom.es](http://www.cedom.es), apartado "Asociados", "Socios EDOM"

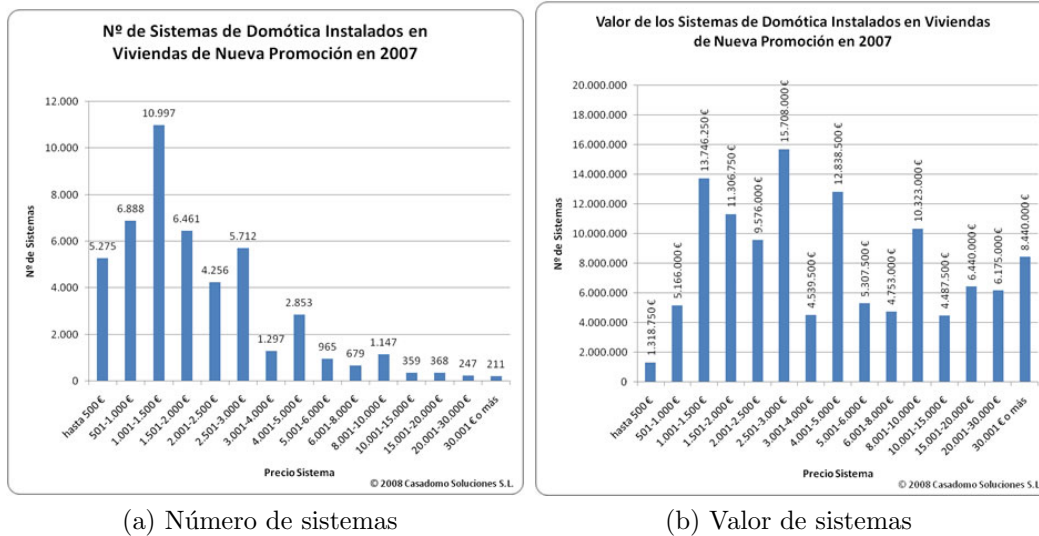


Figura 2.11: Sistemas domóticos instalados en 2007 en viviendas de nueva promoción

En Europa, la cifra del mercado europeo de viviendas inteligentes aumenta, siendo aproximadamente el 74 % en 2012, correspondiente al valor generado por los productos y, el 26 % restante, referente a la integración de sistemas (ver figura 2.12).

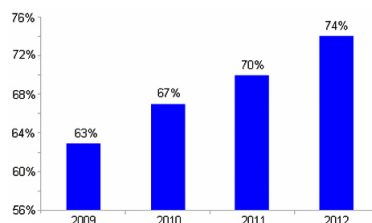


Figura 2.12: Progresión de la cifra de mercado de viviendas inteligentes entre 2009 y 2012.

Así lo revela un estudio llevado a cabo por la consultora británica Bria[14], que incluye varios informes de mercados europeos diferentes.

En dicho estudio se concluye que, el auge de la domótica está relacionado con las soluciones de alta gama. También, que tiene aún espacio para su crecimiento en sistemas más modestos, expandiéndose a un ritmo considerable.

Alemania es el mayor fabricante y consumidor de soluciones inteligentes, representando el 70 % del total de productos fabricados. Países como Francia o Inglaterra, que son los siguientes en la lista, sólo representan el 10 y el 8 por ciento del mercado europeo.

El mercado de alta gama representa dos tercios del mercado total, mientras que el siguiente nicho, que representa un quinto, son las empresas pequeñas y medianas, que usan aplicaciones domóticas en sus instalaciones y edificios (comerciales, hoteles o restaurantes).

Para 2014, CEDOM presenta una serie de objetivos, a raíz de su asamblea general de 3 de junio de 2014, en la que se hizo balance de 2013. De entre todos los proyectos que CEDOM llevó a cabo en 2013, el más destacable fue la elaboración de la propuesta de Documento Reconocido para el Código Técnico para la Edificación (CTE), en la que se recoge una metodología para contabilizar la contribución de la domótica e inmótica en la certificación energética de viviendas y edificios. Esta propuesta se divide en cuatro etapas, basadas en la norma UNE-EN15232[15] “Eficiencia energética de los edificios. Métodos de cálculo de la mejora de la eficiencia energética mediante la aplicación de sistemas integrados de gestión técnica de edificios”. Se trata de un proyecto clave para 2014, ya que CEDOM se dispone a presentarlo y difundirlo en el sector para que sea aprobada por los ministerios de Industria y de Fomento y el IDAE[16].

## 2.2. Realidad aumentada

La Realidad Aumentada (RA), Augmented Reality en inglés, es un campo relativamente reciente. No tiene un mercado grande y la gran mayoría de los trabajos que se desarrollan sobre ella son prototipos, o inicios de tecnologías, con vistas puestas en el futuro cercano.

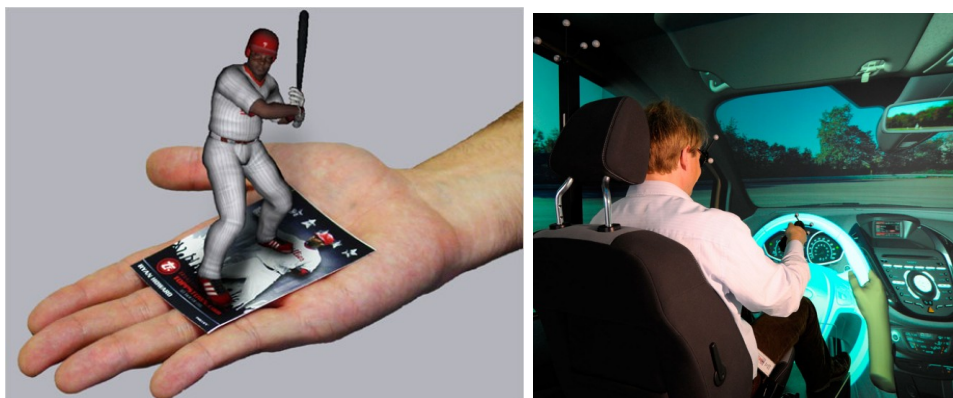
En esta sección veremos aplicaciones reales, dando ejemplos y visualizando las opciones que nos permite la RA.

### 2.2.1. Realidad Aumentada VS Realidad Virtual

La RA y Realidad Virtual (VR) son dos campos cuya línea de separación es bastante borrosa ya que, en cierto sentido, la VR es precursora de la AR donde en algunos aspectos, ambas tecnologías se solapan.

La mayor diferencia está en el uso de una cámara para presentar esa Realidad. En el caso de la VR todo se dispone en formato digital, creando el entorno

en este formato, mientras que la AR superpone una realidad creada, virtual, al propio mundo real que se visualiza a través de la cámara.



(a) Realidad Aumentada: imagen 3D sobre un(cromo real. (b) Realidad Virtual: simulador de conducción.

Figura 2.13: AR vs VR

### 2.2.2. Tipos de RA

A la hora de hablar de RA se debe diferenciar el tipo de entre dos opciones posibles: la basada en posición y la basada en reconocimiento de marcadores u objetos.

Ambos casos superponen a la vista de la cámara una realidad ficticia, pero las condiciones son diferentes ya que la primera hace uso de la información que obtiene basándose en su posicionamiento relativo a un origen de coordenadas, como GPS o su orientación, y la segunda reconoce etiquetas y patrones sabiendo qué debe mostrar sobre ella. Veamos a continuación de qué manera funcionan.

#### RA por posicionamiento

Si bien los smartphones nos ofrecen la opción de hacer uso del GPS y de la orientación cardinal, la Realidad Aumentada basada en posicionamiento hace uso de esta oferta para obtener la información que necesita.

Supongamos que estamos perdidos en una ciudad y queremos saber qué hay a nuestro alrededor. Si cogiéramos un mapa de la ciudad y triangulásemos donde estamos, así como nuestra orientación, podríamos saber qué habrá a nuestro alrededor, siempre limitados por la propia información que muestre el mapa, que irá desde direcciones hasta una breve descripción de los elementos importantes. Es una solución como cualquiera, pero es una solución limitada,



sobre todo, al tamaño del papel. La Realidad Aumentada hace las veces de guía turístico presentándonos, ante la realidad, información de los diferentes elementos que hay a nuestro alrededor. De un sólo vistazo podemos saber en qué entorno estamos, e incluso obtener información de él, simplemente haciendo uso de la interfaz, pudiendo consultar la historia de un monumento o reservar mesa en el restaurante griego que acabamos de descubrir.

El procedimiento general para la AR por geolocalización consta de varias fases:

- Conocer la localización: Con una serie de permisos el software hace uso de los servicios de localización del terminal.
- Buscar lugares de interés: A partir de saber la posición, se busca en una base de datos del servicio los lugares cercanos (se puede configurar el radio de búsqueda). Los lugares que deseen aparecer pueden hacerlo incluyéndose en esta base de datos (como restaurantes, museos o monumentos)
- Descargar información de interés: De estos lugares se descarga la información pertinente al terminal.
- Mostrar la información: Una vez conocida la información se superpone a la visión que ofrece la cámara del dispositivo.

El software que ofrece este tipo de servicio, funciona enfocando el dispositivo alrededor del ambiente mostrando los puntos de interés, informando de su distancia al cliente y permitiendo el despliegue de más información, la cual se puede consultar e interactuar con ella.

Algunos ejemplos son:

- Layar[17]: El más utilizado para aplicaciones de realidad aumentada por geolocalización. Para el desarrollador no hace falta tener excesivo conocimiento de programación ya que la aplicación para desarrollar programas es intuitiva.
- Junaio[18]: Para el desarrollador sólo hace falta una cuenta registrada. Todo el servicio es gratuito.
- Wikitude[19]: Tampoco hace falta conocimientos de programación y su aplicación online es bastante intuitiva.



Figura 2.14: Vista desde un terminal con Wikitude, Layar y Junaio

### RA por reconocimiento de marcadores

El reconocimiento de marcadores se basa en el uso de la cámara del dispositivo, que se orienta hasta que una marca o etiqueta se pueda leer por parte del objetivo: la cámara escanea el contenido y reconoce el valor de la etiqueta, que estará relacionada con un elemento digital que se haya diseñado previamente.

El marker básico sobre el que se trabaja es el de la figura 2.15

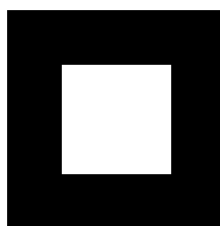


Figura 2.15: Marcador base de AR

a partir de él se desarrollan los marcadores personalizados. Algunos ejemplos de marcas o etiquetas son los de la figura 2.16,

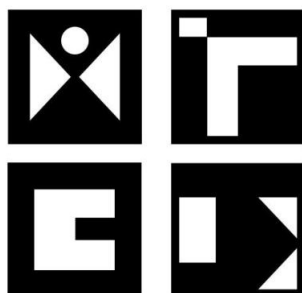


Figura 2.16: Ejemplos de marcadores de AR

también se pueden usar códigos QR[20] o imágenes. Para el uso de estos marcadores hace falta asegurarse de que son correctos, es decir, que combinen todas las especificaciones de dimensiones, definición y distinción de las diferentes partes necesarias para que la cámara la pueda reconocer<sup>2</sup>.

Los marcadores deben cuidar de tener forma cuadrada, los bordes deben contrastar bien y además deben ser de color sólido.

El procedimiento habitual para usar la RA basada en marcadores es:

- Crear el marcador: Para ello cualquier editor, como Scribus, nos puede ayudar a diseñar y dimensionar el marcador. Además hace falta asegurarse de que se trata de una imagen válida. El formato de los marcadores debe ser “.pat”[21].
- Crear las imágenes para mostrar: Con un editor 3D se pueden crear imágenes que se mostrarán sobre la etiqueta.
- Reconocer la etiqueta: Si enfocamos la cámara hacia la etiqueta, el motor de imagen presentará el objeto previamente definido, correspondiente a esa etiqueta.
- Mostrar la imagen: Si en efecto se ha reconocido el marcador, se muestra la imagen superponiéndola a la realidad.

Una visualización típica podría ser la figura 2.17

---

<sup>2</sup>Para asegurarse existe una herramienta online que se puede visitar en <http://flash.tarotaro.org/blog/2009/07/12/mgo2/>



Figura 2.17: Ejemplo de RA sobre marcador

Existen diferentes softwares que ofrecen este tipo de AR, por ejemplo, Vuforia, String o Junaio, aunque existen otras aplicaciones como Aurasma[22], DAQRI[23], o AR-media[24] plugin que se basan en imágenes o códigos QR.

### 2.2.3. Aplicaciones

La Realidad Aumentada nos propone sumergirnos casi literalmente en una nueva vuelta de tuerca de la realidad, pudien interactuar con ella en otra dimensión, más allá de la física, superponiendo imágenes, vídeos o texto, que podemos compartir, leer o cambiar a nuestro parecer. Nuestra realidad engloba cantidad de entornos sobre los cuales la RA puede trabajar y mostrarnos una nueva versión de esta realidad:

**Textos interactivos:** La industria editorial, la prensa o revistas no sólo sirven ya para ser leídos, también nos pueden ofrecer un abanico extra de información e interacción, incluyendo por ejemplo publicidad o biografías de los autores, así como la opción de visitar foros o páginas sobre el tema en cuestión. Se trata de superponer varios kilómetros de papel sobre un único ejemplar, interactivo, intuitivo y sencillo.



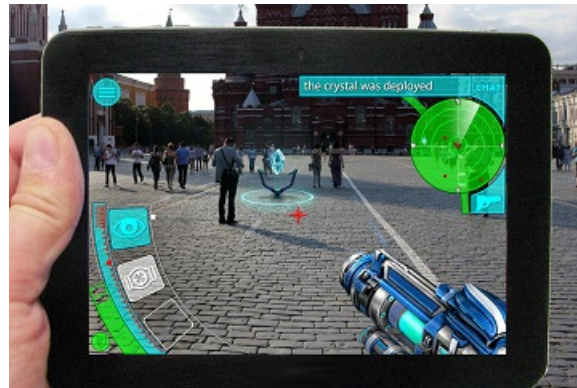
Educación: Hacer más sencilla la comprensión y visualización puede ser esencial para el mejor análisis de la información que ofrece un texto didáctico o una presentación. Presentar en 3D los temas que se ofrecen con el simple gesto de usar el móvil ayuda a entender y a sintetizar la información.



Turismo: Los mapas y la información que necesitemos se unen en una realidad sencilla y útil, justo ante nuestros ojos, sustituyendo mapas o guías pesadas. Esta guía aumentada puede estar hecha por viajeros y autóctonos.



Entretenimiento: La interacción con un mundo real hace más atractiva la oferta ya que se usan objetos reales.



Medicina: Nos da la oportunidad de aventurarnos, antes de hacerlo en la vida real, a realizar una operación o un diagnóstico. Sin ir más lejos, la Agencia Espacial Europea ha creado un dispositivo de realidad aumentada, gracias al cual, los astronautas pueden asistirse a sí mismos[25].



Arquitectura: Nos presenta edificios con las futuras vistas o, incluso proyectar maquetas en 3D sobre un plano, como si ya estuviera construido, o visualizar como era ese edificio en el pasado.



Industria: Del mismo modo, es aplicable el reconocimiento de objetos en sistemas complejos como motores o máquinas, pudiendo conocer su uso o instrucciones, así como su correcto montaje.



Publicidad: añadir una mayor cantidad de datos a formatos como papel o vallas publicitarias ayuda a las agencias a hacer llegar más información a los posibles clientes, así como presentar otro formato más atractivo como vídeos o animaciones.



#### 2.2.4. Actualidad

A día de hoy existen gran cantidad de aplicaciones basadas en la RA para ofrecer algún tipo de servicio en su mayoría de localización o de información.

Veamos algunos ejemplos:

LearnAR[26]: Se trata de un programa de estudio para maestros y alumnos, que ayuda a explorar el mundo real, tanto sobre matemáticas como anatomía, geometría, idiomas o educación física.

WordLends[26]: Permite traducir palabras haciendo una foto sobre el texto, dibujándola traducida sobre la imagen previa.

Wikitude World Browser: Ganador del AugmentedPlanet funciona como una enciclopedia del futuro. Desde cualquier lugar podemos localizar y compartir en redes sociales todo lo que se nos ocurra, como por ejemplo ver los tuits que se generan a nuestro alrededor.

TAT augmented ID[27]: Reconoce la cara de una persona y muestra su información tanto sacada de internet como de redes o haciendo búsqueda en google.

Layar: Es el navegador de realidad aumentada por excelencia. Sea lo que sea lo que quiera localizar, Layar te conduce y te muestra dónde tienes que ir.

Point & Find[28]: Muestra información sobre objetos y se puede compartir la experiencia con otros usuarios.

Junaio: Se trata de un navegador de realidad aumentada que permite sustituir la publicidad por otros contenidos, de esta manera la gente que no esté de acuerdo con la publicidad colocada ante su casa puede hacer, gracias a esta aplicación, que al enfocar sobre ella se visualice otro contenido.

Lookator[29]: Muestra la información de los puntos de anclaje wifi cercanos.

Car Finder[30]: Envía la posición de nuestro coche para más tarde, en el momento de consulta, llevarnos con el mapa y el radar hasta su posición.

Playar[31]: Centrada en publicidad en publicaciones, envases, carteles de cine o de otro tipo.

Google Goggles[32]: Podemos capturar cualquier cosa a nuestro alrededor y se nos muestra información después de un escaneo sobre los objetos que se encuentren, basándose en sus logotipos o textos.

Google Sky Map[33]: Ayuda para el estudio de la astronomía. Basta apuntar en cualquier dirección para que se muestre la posición de las estrellas y constelaciones, así como de la Luna o el Sol.





# Capítulo 3

## Diseño

A la hora de diseñar el prototipo, hace falta saber cómo dividir el proyecto e indentificar los diferentes problemas, como se hizo en el apartado 1.3, e idear un plan para solventarlos, como en el apartado 1.4.

Una vez identificados hace falta tomar decisiones de diseño, pudiendo comparar las diferentes opciones que existen y decantándose por una de estas. En las siguientes páginas de este apartado se exponen las opciones y las decisiones de diseño por las que se ha optado.

### 3.1. Punto de partida.

Antes de concretar las diferentes partes del sistema, hace falta hacer un análisis más en profundidad de sobre cómo se quiere que funcione el sistema.

Se ha comentado que lo que se quiere es interactuar con objetos a través de la cámara, que se muestren las opciones a través de la imagen y que se pueda decidir sobre esta visión que acciones realizar. Para ello, primero se debe reconocer el objeto. A continuación, mostrar en pantalla algún tipo de botón que ayude a decidir sobre qué objeto se quiere interactuar. Una vez decidido, mostrar qué opciones tiene ese objeto, para, más tarde, seleccionar una de esas opciones y ver como se vuelven realidad.

Es decir,

1. Reconocer el objeto a través de la cámara del móvil apuntándola hacia dicho objeto.
2. Mostrar algún tipo de seleccionador por pantalla, un interfaz sencillo compuesto por botones o actuadores similares.

3. Una vez seleccionado el objeto, mostrar las opciones que tiene, para ello hará falta consultar al servidor, interpretar su respuesta y mostrarlas en pantalla.
4. Cuando se haya seleccionado una opción habrá que enviarla al servidor. Éste la interpretará y actuará en consecuencia sobre los objetos para cambiar su estado.
5. Ese cambio de estado se tiene que hacer visible de alguna manera, es decir, se tiene que guardar esa información y se debe mostrar de algún modo.



Figura 3.1: Diagrama del sistema.

Veamos parte por parte.

## 3.2. Reconocimiento de objetos

En el caso de que los propios elementos sepan notificar su presencia existen varios puntos a tener en cuenta:

- En el hogar cada elemento debe llevar un dispositivo, por ejemplo Bluetooth[34], con el coste pertinente.
- Estéticamente cada objeto debe llevar adherido uno de estos elementos, con un tamaño a tener en cuenta.
- El acceso al medio es complicado si el número de objetos es alto.
- El uso no es necesariamente intuitivo, ya que no desarrolla las posibilidades que da el hecho de ver directamente el elemento sobre el que se actúa.

El acceso al medio y la no intuición sobre el sistema son sus dos desventajas mayores.

Por otro lado, modelar un entorno con un sistema de coordenadas requiere un desarrollo y una instalación complejas. A saber:

- Para conocer la posición de cada elemento haría falta desarrollar un sistema de coordenadas.
- Para conocer la posición del terminal móvil sería necesario desarrollar o bien algún tipo de red de sensores o algún otro sistema basado en la potencia de señal recibida.
  - Las redes de sensores requieren un despliegue de medios, como balizas Zigbee[35], que a su vez implica tener un controlador, routers Zigbee y dispositivos finales, así como un estudio pertinente de su topología dependiente de cada entorno.
  - El uso de redes WiFi requiere de un estudio del entorno, y depende de muchos elementos para que sea fiable, ya que las intensidades de las señales fluctúan, así como que los puntos de acceso pueden desaparecer. También es necesario un algoritmo que determine la posición propia y relativa a cada objeto con uso de, por ejemplo, la orientación respecto a las coordenadas cardinales. De nuevo, un estudio demasiado amplio y costoso.

- La complejidad general de este método es demasiado alta, el rendimiento es bajo y la probabilidad de error es alta debido a las fluctuaciones.

El contra a tener en cuenta es su complejidad en la instalación y desarrollo.

El uso de sistemas NFC o RFID requieren:

- Estar suficientemente cerca de cada objeto.
- Que el terminal tenga la opción de uso de este tipo de tecnologías (no todos los terminales tienen módulo NFC).
- Que cada elemento debe incluir un emisor, con su coste y su impacto estético.

El mayor problema es la distancia de actuación sobre los objetos.

El uso de infrarrojos tiene una serie de inconvenientes de la índole de los sistemas NFC o RFID:

- Cada elemento debe llevar un receptor de infrarrojos.
- El terminal móvil debe llevar un módulo de infrarrojos.
- Deja de ser intuitivo su uso, ya que se requieren dos partes a la hora de interactuar: apuntar físicamente al objeto y elegir las opciones dentro del interfaz, lo que puede resultar incómodo.

En el caso de un sistema de infrarrojos su mayor problema es la incomodidad y poca presencia en el mundo de los terminales móviles.

Finalmente la opción de reconocer el objeto a través de marcas tienen otras serie de características a tener en cuenta:

- El reconocimiento se hace a través de la cámara, por lo que debe reunir ciertas características de definición y calidad.
- El objeto debe incluir un marcador, adherido a su superficie visible, afectando a su estética.
- Se debe tener visión directa con esta marca.
- El uso de la cámara hace intuitivo su uso.

- En el momento en el que avancen los sistemas de reconocimiento de objetos, no harán falta añadir marcas a los objetos, reconociendo directamente el objeto en sí.

Para poder decidir cuál de estos sistemas debemos basarnos en qué queremos que reuna nuestro sistema.

- Que el sistema sea intuitivo a través del uso de la cámara del terminal.
- Que no haga falta estar necesariamente cerca del objeto.
- Que el elemento sobre el que queremos actuar no lleve nada añadido, o que, si debe hacerlo, sea lo menos impactante y lo más barato posible.
- Que la complejidad de instalación del sistema sea lo más baja posible.

Basándonos en las opciones descritas y qué características debe tener nuestra solución, la mejor opción es el uso de marcadores/pegatinas con vista a que en un futuro siquiera hagan falta, ya que los objetos se reconocerán directamente.

### 3.3. Realidad Aumentada

Una vez sabemos cómo identificar el objeto hace falta mostrar al usuario algún tipo de seleccionador, ya sea o bien un recuadro alrededor del objeto, o un simple botón que se superponga a la imagen, es decir, hacer uso de la Realidad Aumentada.

Existen varias empresas y desarrolladores de RA de acceso libre, de fácil diseño e instalación como Vuforia, Junaio o AndAR de Google. En este caso se ha decidido usar la librería de Google, ya que su uso es bastante más sencillo y, para las necesidades que se deben satisfacer en este proyecto, se trata de un recurso más que suficiente que las cubre. Además, el sistema operativo del terminal será Android, por varias razones:

- Es de código abierto.
- Existe un gran número de bibliotecas que aportan sencillez al diseño, ya que se trata de procesos ya desarrollados y publicados.
- Además hay una gran cantidad de foros de discusión donde poder formular preguntas y encontrar respuestas.

- Se trata de un sistema operativo de gran presencia a escala mundial.

Este interfaz de usuario será por tanto una superposición de elementos tipo Button de la librería Android que seguirá a la marca que representa al objeto con el que se quiere interactuar.

### 3.4. Cliente

Una vez seleccionado el objeto con el que actuar, se debe hacer una petición al servidor por parte de un cliente.

El cliente realizará las peticiones pertinentes al servidor en formato HTTP[36], ya que existe una biblioteca de fácil uso en Android, es un protocolo sencillo y no suele tener problemas de firewalls.

A la hora de pedir información, se hará con el siguiente formato: “http: // direccionIP : puerto / nombreAplicacion / carpeta1 / ... / carpetaN / objeto”, respondiendo el servidor con el contenido de ese documento, que contiene una representación virtual del objeto de la petición.

Cuando el servidor responda, el cliente interpreta esa respuesta, mostrando por pantalla las opciones de ese objeto. Cuando se seleccione una opción debe enviarse al servidor para que sea ejecutada. Ésto lo hará a través de una comunicación directa con el simulador que representa un mundo virtual sobre el que actuar. Para ello se ha decidido usar Sockets[37] ya que, de nuevo existen librerías para su manejo y una gran cantidad de foros de donde sacar respuestas, así como su uso a lo largo de los estudios que han precedido este trabajo fin de grado.

El formato de este mensaje será tal que “nombreObjeto/idopcion”.

### 3.5. Servidor

En el planteamiento del problema se han sugerido dos opciones a la hora de guardar la información de cada objeto. A modo de prueba de concepto, se ha optado por un sistema en el que un servidor guarde la información de estado, así como que sea, al mismo tiempo, el actuador sobre los objetos que se disponen en este trabajo, a través de una interfaz gráfica, un simulador en el que se muestra un mundo virtual controlable.

A la hora de posicionarse ante el tipo de servidor que se va a usar es necesario definir criterios de elección:

- Debe ser de acceso abierto.
- Debe ser de fácil uso.
- Debe hacerse cargo de las peticiones y respuestas de forma automática.
- Debe usar pocos recursos.

Basándonos en este criterio se usará un servidor Apache Tomcat[38], ya que cuenta con una amplia documentación y se ha estudiado a lo largo de la carrera, con lo que su implantación sería sencilla y no añadiría un tiempo de aprendizaje a la hora de desarrollar este proyecto.

El servidor responderá a las peticiones que se hagan por parte del cliente a través de un sistema de ficheros en el que se guardará la información de estado de cada objeto.

Debe existir por lo tanto alguna forma de cambiar el contenido de estos, de sobreescribirlos.

### 3.5.1. Estructura de ficheros

Los objetos, dependiendo del tipo que sean, tendrán una información que guardar referente a su estado. Toda esta información se guardará en formato JSON[39]. Para la prueba de concepto se modelarán dos clases de objetos:

1. Tipo ONOFF: Serán aquellos que sólo tengan dos posiciones: encendido y apagado o apertura y cierre.
2. Tipo MODULABLE: Serán aquellos que tengan posiciones intermedias entre encendido y apagado, representando su posición en porcentajes correspondiendo “0” a totalmente apagado y “100” a totalmente encendido.

Un programa que corre en el servidor interpretará las acciones enviadas por parte del cliente y realizará los cambios tanto en el fichero del objeto como en un mundo virtual que presentará un posible escenario cotidiano, mostrando el estado de los objetos de manera visual (véase apartado 3.7).



### 3.6. Base de datos

Como ya se ha comentado será necesario guardar de alguna manera distintos aspectos del sistemas: qué objetos hay y qué opciones, para actuar sobre él, tiene.

En este caso se ha usado una sistema MySQL[40], por varias razones:

- Se trata de una de las bases de datos de código abierto más usada para Web.
- Su uso es compatible en diferentes plataformas.
- Es escalable.
- Es compatible con protocolos seguros.
- Facilidad de configuración e instalación.
- Robusto soporte transaccional (ACID[41]).
- Existen bibliotecas para su control.
- Existe una gran documentación al respecto, tanto profesional como dada por desarrolladores autónomos.
- Al ser de un uso tan extendido existen cantidad de foros específicos para consulta de dudas.

En el caso de MySQL, los datos se guardan a través de tablas, que contienen diferentes columnas que se pueden definir, así como un nombre de la propia tabla. La figura 3.2 un ejemplo de este tipo de tablas.

```

C:\Windows\system32\cmd.exe - mysql -u root -p -h 127.0.0.1
mysql> null;
Query OK, 0 rows affected (0.12 sec)

mysql> show tables;
+-----+
| Tables_in_userlogin |
+-----+
| user                 |
+-----+
1 row in set (0.00 sec)

mysql> describe tables;
ERROR 1146 (42S02): Table 'userlogin.tables' doesn't exist
mysql> describe user;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id     | int(11) unsigned | NO   | PRI | NULL    | auto_increment |
| username | varchar(25)      | NO   |     | NULL    |                |
| passwd | varchar(25)      | NO   |     | NULL    |                |
| email  | varchar(40)      | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>

```

Figura 3.2: Ejemplo tabla MySQL

Las tablas guardarán información sobre los objetos que hay y sobre qué opciones tienen. Estas serán consultadas con las diferentes peticiones del cliente.

Para hacer de su uso algo más sencillo, a la hora de ampliar el número de objetos sobre el que actuar, se desarrollará un programa que ayude a manejar la base de datos de una forma sencilla.

### 3.7. Simulador: un mundo virtual

Ya que no se va a desarrollar un sistema de actuadores físicos, dejándose para ampliaciones posteriores, se ha decidido llevar a cabo una pequeña y sencilla interfaz gráfica que represente de forma visual el estado de los objetos.

### 3.8. Restricciones y marco regulador

La filosofía de este proyecto se basa en el principio de Código Abierto, Open Source. Esta expresión es con la que se conoce al software que es distribuido y desarrollado libremente, focalizándose en los beneficios prácticos.

El lado positivo de tener acceso al código fuente, a usarlo, reescribirlo, modificarlo y distribuirlo es, que, potencialmente, cualquiera puede ser un desarrollador. Las soluciones privadas hacen que un proyecto esté acotado por el número de personas que trabajan sobre él, siendo incluso un secreto hasta el

momento en el que el producto está listo.

En el caso de este proyecto, todo el sistema se ha desarrollado bajo la licencia de Google, usando tanto Android como la librería AndAR y la librería GSON, además de usar las licencias de Apache Tomcat. Veamos a continuación en qué consisten éstas y qué restricciones tienen.

En cuanto a Google caben destacar los siguientes puntos de su Términos y Condiciones<sup>1</sup> entre otros:

- Android pertenece al proyecto de Android “Android Open Source Project” [42].
- Ud. no debe usar el SDK[43]. y no debe aceptar los Términos y Condiciones si es una persona a la que se le ha obstaculizado recibir el SDK bajo las leyes de los Estados Unidos u otros países, incluyendo su país de residencia o aquel desde el que usa el SDK.
- Sujeto a los términos de esta licencia, Google garantiza una licencia limitada, mundial, libre de pagos, no atribuible, no exclusiva, para hacer uso de su SDK solamente para desarrollar aplicaciones para ser usadas en plataforma Android.
- Está de acuerdo que la forma y naturaleza del SDK que Google le provisiona puede cambiar sin aviso previo y que las futuras versiones del SDK pueden ser incompatibles con versiones anteriores de aplicaciones desarrolladas con anteriores SDK.
- Está de acuerdo en usar el SDK y escribir aplicaciones sólomente con propósitos que sean permitidos bajo esta licencia y leyes aplicables o regulaciones.
- Está de acuerdo que si Ud. usa el SDK para desarrollar aplicaciones para usos publico de usuarios, protegerá la privacidad y los derechos legales de estos usuarios.
- Está de acuerdo con que sólomente Ud. es responsable de cualquier infracción de sus obligaciones bajo este acuerdo de licencia.
- Con idea de continuar innovando y promoviendo el SDK, Google coleccionará ciertas estadísticas de uso del software incluyendo, pero no sólomente, direcciones IP, versión del software e información de que herramientas y/o servicios del SDK se usan y de qué manera. Antes de

---

<sup>1</sup><https://developer.android.com/sdk/terms.html>

que esta información se recoja, se notificará a través del SDK pudiendo hacer que no se recolecte en el futuro.

- Si usa aplicaciones desarrolladas por terceros, Google no se hace responsable de esas aplicaciones, datos, contenidos o recursos y que no está sujeto a responder de ningún daño o pérdida que se pueda derivar de esto.

En términos generales Google pone a disposición de los desarrolladores sus librerías, siempre y cuando las aplicaciones creadas se mantengan dentro de la legalidad.

En cuanto al servidor Apache Tomcat existe una licencia<sup>2</sup> de uso de la cual caben destacar el siguiente punto:

- Por la presente se le garantiza un perpetuo, mundial, no atribuible, no exclusivo, libre de pagos y cargos, e irrevocable derecho de reproducir, realizar trabajos derivados y distribuir este trabajo.

El resto de documentación no incumbe a este proyecto, ya que se trata el uso de la marca o la aceptación de esta licencia.

Por otra parte, Scribus, programa con el que se han realizado los marcadores, está también sujeto a la iniciática Open Source al igual que Eclipse.

De esta manera, tanto los términos de las diferentes partes, como la Ley Orgánica de Protección de Datos forman el marco regulador del presente proyecto, así como la norma UNE-EN 15232 del apartado 2.1.3.

---

<sup>2</sup><http://www.apache.org/licenses/LICENSE-2.0>



## Capítulo 4

# Trabajo realizado: arquitectura del prototipo

En los siguientes apartados se describe la arquitectura del prototipo desarrollado, desgranando las diferentes partes que lo componen, los recursos que se usan y sus estructuras.

La aplicación final resulta ser un lector de marcadores los cuales tienen relacionados un objeto (uno a uno), sobre los cuales se puede actuar a través de una interfaz gráfica. El diagrama de flujo que sigue tiene los siguientes pasos:

1. Reconocer el marcador.
2. Mostrar botones para que el usuario pueda seleccionar un objeto.
3. Pedir las opciones al servidor.
4. Mostrar las opciones con las que el servidor ha respondido.
5. Enviar al servidor la opción elegida por el usuario.
6. El servidor realiza la acción cambiando el simulador del mundo real.

### 4.1. Arquitectura

En la figura 4.1 se diferencian las partes que componen el total de este proyecto. A saber:

- Cliente.
  - Terminal

#### 44CAPÍTULO 4. TRABAJO REALIZADO: ARQUITECTURA DEL PROTOTIPO

- Realidad Aumentada
  - Interfaz de usuario
  - Sistema de reconocimiento de marcadores.
- Servidor
    - Apache Tomcat
    - Sockets
    - Base de datos
    - Sistema de ficheros
    - Simulador

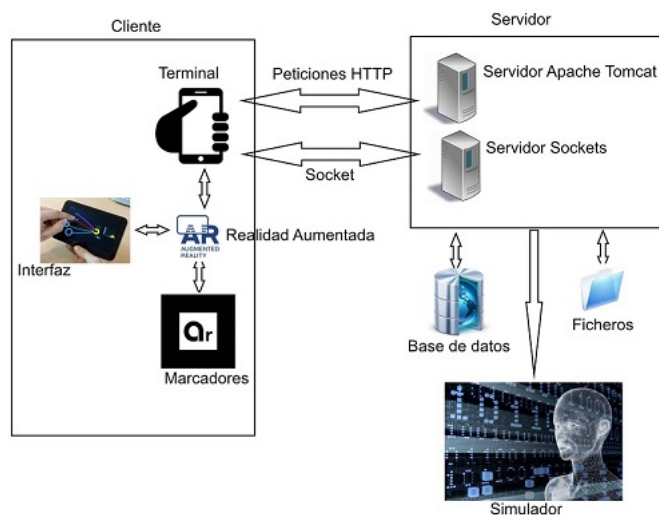


Figura 4.1: Arquitectura del proyecto.

En este trabajo hace falta diferenciar dos partes claras: un cliente y un servidor. La necesidad de dividir este sistema de esta manera es, que por un lado el cliente tendrá corriendo dos tipos de procesos: uno para realizar peticiones sobre información de objetos, y otro que realizará el diálogo con el usuario a través de un interfaz gráfico; y por otro, que el servidor será el encargado de guardar toda la información referente a los objetos y además mostrará su cambio de estado en un mundo virtual creado a tal fin. La arquitectura del servidor se divide tanto en un servidor Apache Tomcat como en un programa de control de ficheros, donde se guarda la información de

estado.

El servidor Apache Tomcat es un contenedor web, basado en el lenguaje java que actúa como motor de servlets[44] y JSPs[45].

Su arquitectura se puede ver en la figura 4.2

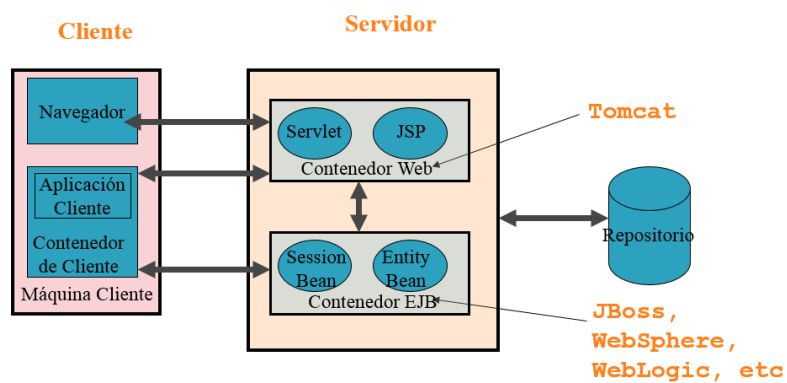


Figura 4.2: Funcionamiento del servidor Apache Tomcat.

Por otro lado la estructura de ficheros es: “/webapps/FAR/”<sup>1</sup>, dónde se almacena el estado de los objetos.

Las figuras 4.3 y 4.4 muestran cómo es guardado el estado de un objeto.



Figura 4.3: Contenido de la carpeta “webapps”.

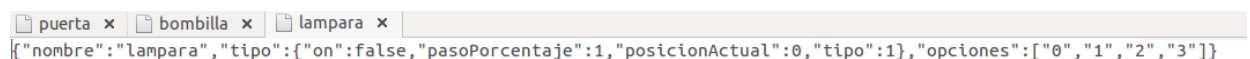


Figura 4.4: Contenido del archivo “lampara”.

<sup>1</sup>El nombre del directorio viene del acrónimo de For Augmented Reality, elegido por el autor



## 4.2. Descripción funcional

### 4.2.1. Cliente

Empecemos por el cliente. Se ha comentado que existe un motor de realidad aumentada. Este proceso se divide en dos funciones claras: presentar la información al usuario e interpretar los marcadores. Veamos estas partes:

1. Reconocimiento de los marcadores: Se encargará de reconocer los marcadores y saber qué objeto corresponde a qué marcador, para ello se hace uso de la biblioteca AndAR de Google.
2. Interfaz de usuario: Para que el usuario pueda interactuar con el entorno se debe presentar de forma intuitiva las posibilidades que existen tanto en el número de objetos como en sus acciones.

En cuanto al control de la realidad aumentada de esta aplicación, cabe destacar tres clases principales, más una cuarta que sirve de herramienta para posicionar los botones de acción sobre los objetos. Su función se describe a lo largo de los siguientes apartados.

Por otro lado, el terminal debe establecer un intercambio de información con el servidor. Para ello existen dos tipos de comunicaciones: basada en peticiones HTTP y en el establecimiento de un Socket entre ambos. La primera sirve para conocer las opciones que guarda el objeto, mientras que la segunda envía la elección del usuario, para que el servidor proceda a realizar los cambios de estado sobre los objetos.

En el apartado 4.3 se describe cómo funcionan cada una de estas partes.

### 4.2.2. Servidor

El servidor tiene dos tareas: responder a las peticiones HTTP e interpretar los mensajes que envía el cliente a través del Socket.

La primera tarea es la más sencilla, ya que el contenedor Apache Tomcat las realiza de manera automática. Sin embargo, la interpretación de las órdenes implica varias funciones:

1. Leer el mensaje del socket.
2. Sacar información de este mensaje para saber qué hacer y sobre qué objeto.

3. Leer la base de datos para conocer los objetos sobre los que tiene control
4. Leer y escribir los ficheros que contienen información de estado, tanto para conocerla como para actualizarla.
5. Realizar los cambios de estado mandando órdenes a actuadores. En este caso, un simulador.

En el apartado 4.3 se describe cómo funcionan cada una de estas partes.

## 4.3. Descripción operacional

### 4.3.1. Cliente

#### **Realidad Aumentada: Reconocimiento de marcadores**

La idea del reconocimiento de marcadores es asociar un marcador a un objeto, es decir, tiene que haber un momento en el que se diga al sistema que el “marcador1” corresponder al “objeto1”. ¿Cómo se hace esto?

El sistema principal es una clase llamada Activity, que hereda de AndARActivity, una clase la biblioteca AndAR, que a su vez hereda de Activity, propia de Android, e implementa interfaces de control de la cámara.

La clase Activity, propia de Android, es una clase que construye la vista de las diferentes pantallas de navegación que contiene una aplicación Android. Es decir, cada pantalla diferente que se muestra al usuario es un objeto Activity. Éste, tiene asociados una serie de métodos que eventualmente son llamados para iniciar su vista, o saber qué hacer cuando se pasa a segundo plano.

Los métodos más importantes de los que uno tiene que hacerse cargo a la hora de crear una nueva vista son:

1. onCreate(Bundle): Que se llama de manera automática por el sistema y sirve para iniciar la vista. Típicamente en este método se configuran e inician las variables y se añaden vistas.
2. onPause(): Se llama también de manera automática cuando el usuario pasa esa ventana a un segundo plano, ya sea al navegar por la propia aplicación o al iniciar otras.

En la figura 4.5 se puede ver el ciclo de vida de un objeto Activity de Android.

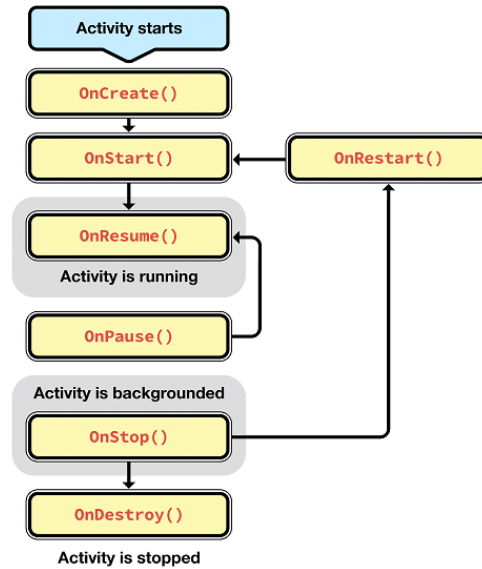


Figura 4.5: Ciclo de vida de un objeto Activity de Android.

Nuestra clase Activity contiene dos Atributos principales del tipo ARObject [46] y ARToolkit[47], ambos también de la biblioteca de Google. La función de cada uno de estos atributos es la de representar objetos que tienen relacionados un marcador (es decir, por cada objeto del mundo real existe un marcador relacionado. Este par se representa con un objeto de esta clase), y por otro lado la de controlar todo el sistema, ya que ARToolkit proporciona una serie de funciones para la captura de vídeo y para la búsqueda de ciertos patrones, así como para el registro de esos objetos ARObject.

Al iniciar la actividad el sistema registra en ARToolkit los diferentes objetos. En nuestro caso, cada uno de estos objetos ARObject son hijos de la misma, llamados CustomObject1, a los cuales se les han añadido métodos y atributos. El modo de funcionamiento es el siguiente:

1. Inicio de la aplicación: La aplicación arranca y realiza una serie de configuraciones:
  - a) Crea un objeto de la clase CustomRenderer que implementa a la interface OpenGLRenderer.



Cabe destacar que en la esquina superior izquierda se puede ver un pequeño candado de color rojo el cual indica que la aplicación está bloqueada. Para desbloquear sólo hay mantener pulsado un segundo en cualquier parte de la pantalla.

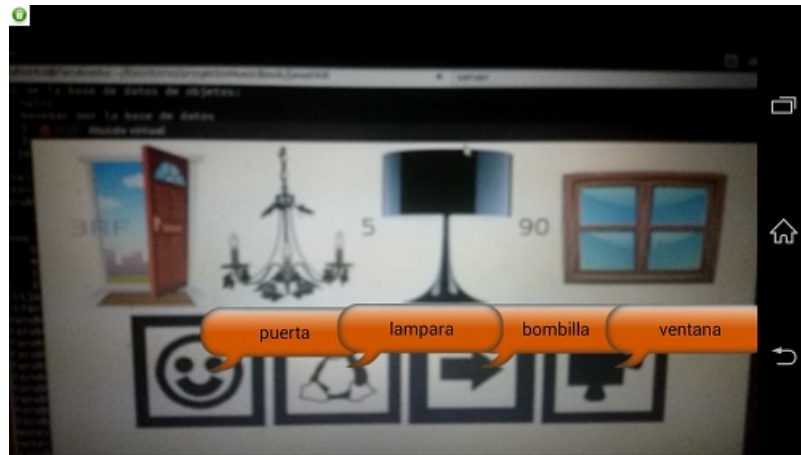
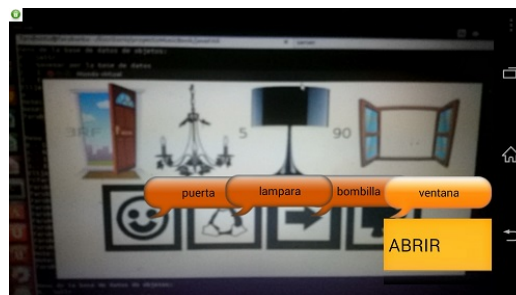


Figura 4.7: Vista del interfaz de usuario con marcadores.

De esta manera, se desbloquea y el icono se vuelve de color verde.



(a) Opción no seleccionada



(b) Opción seleccionada

Figura 4.8: Vista del interfaz de usuario con lista de opciones.

### 4.3.2. Servidor

El servidor se ha dividido en dos módulos: uno encargado de las peticiones HTTP que tienen como fin conseguir las opciones guardadas de un objeto y, otro dedicado a interpretar las órdenes dadas desde el terminal.

#### **Domótica: Servidor HTTP**

En el caso del servidor HTTP, se usa un contenedor web Apache Tomcat el cual responde a las peticiones que se hagan desde el cliente con el contenido de los archivos correspondientes a los diferentes objetos (veánse las figuras 4.3 y 4.4).

#### **Domótica: Servidor de control**

El esquema de las clases java que controlan el servidor y el estado de los objetos es el que se ve en la figura 4.9

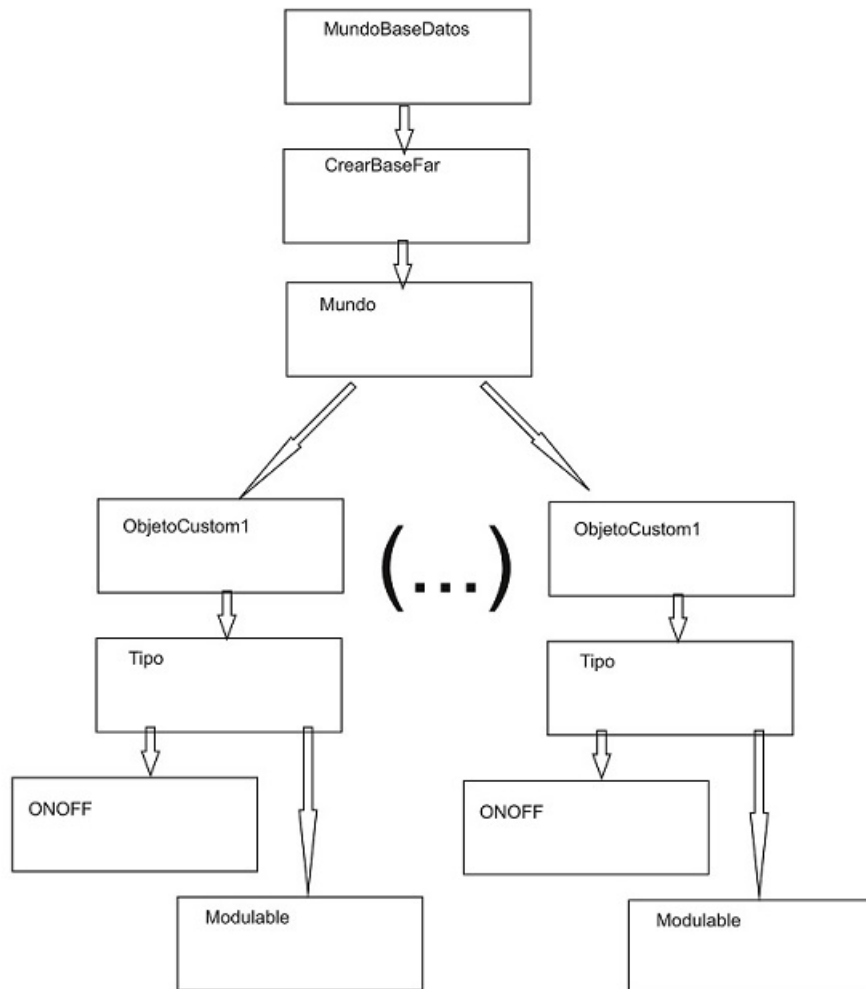


Figura 4.9: Diagrama de clases java del servidor.

La clase Mundo.java se encarga de controlar el socket de donde extrae los datos referentes al nombre del objeto y la opción seleccionada.

Además, como ya se ha comentado, hace las veces de actuador, manifestando en un mundo virtual, un Frame, los cambios que se derivan de las acciones que el cliente hace. En este caso, se han colocado los marcadores pertinentes junto a los objetos para hacer una vista más atractiva, como se puede ver en la figura 4.10.

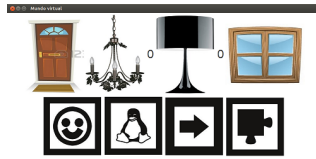


Figura 4.10: Vista del Mundo Virtual.

En el momento en el que recibe una opción, que responde a uno de los códigos 0, 1, 2 ó 3, ejecuta esa opción dependiendo del nombre del objeto. El método `ejecutarOpcion()` recibe el mensaje íntegro del Socket, en él se puede comprobar el nombre del objeto y la opción elegida. Dependiendo del objeto se actúa sobre las imágenes mostradas en el Frame[48].

#### **Domótica: Base de datos y ficheros.**

La clase “`CrearBaseFar.java`” es la encargada de controlar el registro en la base de datos, creando las tablas y ficheros, manteniéndolos actualizados cuando desde la clase `MenuBaseDatos.java` se requiera. Hace uso de la biblioteca `java.sql` para ejecutar los comandos de una manera más sencilla.

Los atributos de esta clase guardan información sobre la base de datos y su acceso, así como el directorio de archivos (dentro de la carpeta “webapps”), y la herramienta JSON, que es usada cuando se requiere escribir en el fichero de un objeto.

Por otro lado la clase `MenuBaseDatos.java` controla el estado de los objetos, así como la representación del Mundo virtual.

Esta clase se encarga de iniciar todos los servicios del servidor. En primer lugar se encarga de tener un acceso a la base de datos intuitivo, de forma que iniciado este programa, despliega un menú de control con el que se puede actuar sobre la base de datos (figura 4.11), eliminando objetos, añadiendo o consultándolos.

También se encarga de iniciar el simulador `Mundo.java`.



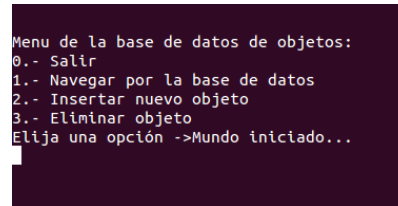


Figura 4.11: Vista del menú de control de la base de datos.

## 4.4. Implementación

### 4.4.1. Cliente

El cliente es el encargado de hacer las peticiones pertinentes al servidor. En total hará dos: una para saber qué opciones tiene un objeto y otra para enviar la opción elegida.

Para saber cómo funciona, hace falta saber primero con qué clases contamos dentro del cliente, es decir, de la aplicación Android que corre en el terminal.

Existe una clase que controla toda la actividad de la aplicación llamada “Activity”. Sin embargo, cabe resaltar el hecho de que se encarga de registrar los objetos que se sabe que existen, añadiéndoles el marcador pertinente. Cada objeto de éstos se trata de un objeto de la clase CustomObject1.java, que controla su detección además de hacer que se visualicen los controles de los objetos. Lo reseñable de esta clase, a la hora de hacer las peticiones HTTP, es una clase interna, que corre en background.

Una vez obtenida la respuesta, se crea la lista de opciones con el método “crearListaDeOpciones(String respuesta)”.

Cuando la lista es creada se despliega y se muestra por pantalla. En el momento en el que el usuario seleccione una de estas opciones de la lista, se manda un segundo mensaje con su formato pertinente, a través de un Socket con el Servidor. Para ello, hay un segundo hilo que, con la misma estructura que el anterior produce el intercambio de información, es decir, corre en Background de manera asíncrona.

En el caso de la petición de opciones ocurre el siguiente intercambio:

1. El cliente envía una petición HTTP a la dirección “http://” + serve-

rAddress + ":8080 / FAR / " + objetoPetición.

2. El servidor Apache Tomcat responde con el contenido del fichero (un objeto JSON)
3. El cliente interpreta este mensaje y lo convierte a un objeto tipo Objeto.java. De esta manera el cliente tiene control sobre este objeto.

Una vez desplegada la lista, el cliente tiene dos opciones, o bien querer interactuar sobre otro elemento, con lo que de nuevo ocurriría el proceso anterior, o bien, querer ejecutar una de las opciones que se le muestran por pantalla. En este caso el intercambio de mensajes se hace a través de un socket con el siguiente orden:

1. Se establece la conexión, se abre un Socket.
2. Se escribe en el canal de salida hacia el servidor.
3. Se cierra el canal y el Socket.
4. El servidor recibe el mensaje con el siguiente formato: "nombreObjeto/identificadorOpcion".
5. El servidor saca la información relevante, el nombre del objeto sobre el que se quiere actuar y la acción elegida.
6. El servidor ejecuta la acción sobre el objeto.

#### 4.4.2. Servidor

Del lado del servidor, es necesario subrayar varias implementaciones.

A la hora de guardar un objeto en la estructura de ficheros, hay que tener en cuenta el tipo del objeto<sup>2</sup>, guardando una información diferente para cada tipo (el estado es una representación JSON de los objetos):

- Para objetos de dos posiciones (TipoONOFF):
  - El nombre del archivo será el nombre del objeto (e.g.: "puerta")
  - El contenido de este archivo será:  
"nombre":"puerta", "tipo":"on":false, "tipo":0, "opciones":["0", "1"].

---

<sup>2</sup>En esta prueba de concepto se ha querido establecer solamente dos tipos.

- Para objetos con posiciones intermedias (TipoMODULABLE):
  - El nombre del archivo será el nombre del objeto (e.g.: “bombilla”)
  - El contenido de este archivo será:  
   “nombre” : “ bombilla ” , “ tipo ” : “ on ” : false, “pasoPorcentaje”  
   : 1, “posicionActual” : 0  
   , “tipo” : 1 , “opciones” : [ “ 0 ” , “ 1 ” , “ 2 ” , “ 3 ” ] .

El tipo de los objetos su tipo, teniendo valor “1” en caso de ser MODULABLE y a “0” en el caso de ser ONOFF. Estas dos clases guardan la información relevante al estado del objeto.

En el caso del TipoONOFF simplemente se indica si está o no encendido ya que no tiene posición intermedia, mientras que en el caso del TipoModulable hace falta guardar:

1. Una variable que indica que esta apagado o es cualquier otro caso.
2. Un valor que, a la hora de subir o bajar la intensidad, usará para dar pasos de ese valor porcentual.
3. Y por supuesto la posicion en la que se encuentra la intensidad del objeto.

Es posible tener varios objetos. Cada uno de ellos contendrá un tipo (Tipo.java) que puede ser o bien de dos posiciones, TipoONOFF.java o de, además, puntos intermedios, TipoMODULABLE.java.

El estado de los objetos reales se guarda en una clase llamada “Objeto.java”, que guarda el nombre del objeto, de que tipo se trata (ONOFF o MODULABLE) y las diferentes opciones que tiene el objeto (Abrir, Cerrar, Subir o Bajar).

Del servidor Apache es necesario destacar cuatro directorios:

1. /bin: En él se encuentran scripts de útiles para Unix y Windows, así como los de arranque y detención del servidor (startup.sh y shutdown.sh, respectivamente.)
2. /conf: Ficheros de configuración.
3. /logs: Ficheros de log, es decir de control de mensajes y estado del servidor.

4. /webapps: Directorio de despliegue para las aplicaciones web. En este se encuentran los archivos de este proyecto.

La clase Mundo.java se encarga de controlar los sockets. Se trata de una clase que implementa la interface Runnable[49], con lo que se trata de un hilo que arranca en el momento en el que se inicia el control de la base de datos (MenuBaseDatos.java).

Además, la base de datos mantiene un registro de los objetos sobre los que se puede actuar, así como sus opciones.

La base de datos contiene una tabla por cada objeto así como un registro de objetos en una tabla llamada IDandOBJECTS, como se puede ver en la figura 4.12.

```
mysql> show tables;
+-----+
| Tables_in_far |
+-----+
| IDandOBJECTS |
| bombilla     |
| lampara      |
| puerta       |
| ventana      |
+-----+
5 rows in set (0.00 sec)
```

Figura 4.12: Vista de las tablas que existen en la base de datos.

La tabla que mantiene el registro de todos los objetos, IDandOBJECTS, tiene la siguiente estructura:

```
mysql> describe IDandOBJECTS;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(25)   | NO   | PRI | NULL    | auto_increment |
| nombre_objeto | varchar(25) | YES  |     | NULL    |               |
| tipo       | int(11)   | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Figura 4.13: Descripción de la tabla IDandOBJECTS.

y el contenido de ésta es:

```
mysql> select * from IDandOBJECTS;
+-----+-----+-----+
| id | nombre_objeto | tipo |
+-----+-----+-----+
| 54 | bombilla      | 1    |
| 57 | ventana       | 0    |
| 58 | puerta        | 0    |
| 59 | lampara       | 1    |
+-----+-----+-----+
4 rows in set (0.01 sec)
```

Figura 4.14: Vista del contenido de la tabla IDandOBJECTS.

Como se ha dicho, cada objeto tiene una representación en la base de datos en la que se guardan sus opciones. A modo de ejemplo, las siguientes figuras, 4.15 y 4.16 describen la tabla del objeto con el nombre “lampara” y su contenido, respectivamente:

```
mysql> describe lampara;
```

| Field         | Type    | Null | Key | Default | Extra          |
|---------------|---------|------|-----|---------|----------------|
| id_opcion     | int(25) | NO   | PRI | NULL    | auto_increment |
| nombre_opcion | int(25) | YES  |     | NULL    |                |

2 rows in set (0.00 sec)

Figura 4.15: Vista de la descripción de la tabla “lampara”.

```
mysql> select * from lampara;
```

| id_opcion | nombre_opcion |
|-----------|---------------|
| 1         | 0             |
| 2         | 1             |
| 3         | 2             |
| 4         | 3             |

4 rows in set (0.00 sec)

Figura 4.16: Vista del contenido de la tabla “lampara”.

## 4.5. Escenarios de pruebas

Cronológicamente, se desarrolló en primer lugar el módulo correspondiente al control domótico y, a continuación, el módulo sobre realidad aumentada para concluir con el ensamblaje total. Las pruebas que se han ido haciendo seguían este orden.

En el caso del control domótico se llevó a cabo pruebas con uno o varios usuarios. Los posibles errores podían ser:

- Respuesta del servidor Apache: al haber varias peticiones, cabe la posibilidad de que no pueda reaccionar a alguna. El escenario en el que nos movemos implica pocos terminales haciendo peticiones.
- Respuesta del programa MenuBaseDatos.java: surge de la misma preocupación que hubo sobre la respuesta del servidor. En este caso no hubo ningún problema sobre el control ya que el número de peticiones es considerablemente reducido.
- Sobreescritura de los archivos dentro del directorio “webapps”: ya que se controla la lectura y escritura desde un programa, que no corre como

un script dentro del servidor, cabía la posibilidad de que no se pudieran sobrescribir los archivos.

Por otro lado, la parte que se encarga del sistema de realidad aumentada podía tener los siguientes:

- Reconocimiento de los marcadores: la lectura de los marcadores depende de la cantidad de éstos que se vea, dependiendo del ángulo y/o superficie. En ciertas circunstancias, como ángulos demasiado abiertos, pueden surgir brillos que dificulten la distinción de los marcadores.
- Intercambio de mensajes: en el momento en el que se abren las peticiones por parte del cliente hace falta controlar el cierre de las mismas para no saturar al terminal ni a la red.

La aplicación final dependía del ensamblaje de estas dos partes. El método que se siguió fue el de incluir, dentro del código de reconocimiento de marcadores, el del cliente de pruebas del sistema de control domótico, probando cada nuevo módulo uno a uno hasta el ensamblaje total.

| Escenario  | Descripción  |
|--|--|
| Respuesta del servidor                             | Varios clientes piden información al servidor                      |
| Programa de control de la base de datos            | Añadir, eliminar y ver la información guardada en la base de datos |
| Escritura de los archivos del directorio "webapps" | Escribir, leer y actualizar los ficheros de los objetos            |
| Reconocimiento de marcadores                       | Lectura e identificación de diferentes marcadores                  |
| Intercambio de mensajes                            | Intercambio de información entre cliente y servidor                |
| Aplicación final                                   | Correcto funcionamiento del sistema completo                       |

Cuadro 4.1: Escenarios de pruebas.

La tabla 4.1 recoge el total de las pruebas realizadas.



# Capítulo 5

## Historia del proyecto

En las siguientes secciones se hace un repaso de cómo se ha llevado a cabo este trabajo, desde el planteamiento inicial hasta el resultado final. Además se añade un presupuesto posible

### 5.1. Fases de desarrollo

En el desarrollo de este trabajo se han pasado por varios puntos de inflexión, relacionados tanto con la posibilidad de compatibilizar el estudio de últimas asignaturas así como empleos no relacionados con este campo de estudio.

**Primera fase: la idea** En Enero de 2013 se propuso la idea de crear un sistema que reconociera los objetos a través de la cámara al que más tarde iba a ser el tutor de este trabajo.

La idea surge de un momento de reflexión personal en el que me preguntaba si existiría alguna forma de hacer un mando universal y sencillo (empujado por un artículo sobre el Internet de las Cosas).

**Segunda fase: primera conversación** A las pocas semanas se mantuvo una conversación sobre el tema, en la que se sugirieron ideas a vuela pluma, ya que se trataba de un proyecto que no iba a empezar hasta el siguiente año académico.

**Tercera fase: intercambio de correos** Al comienzo del año académico 2013-2014 se intercambiaron correos entre el tutor y el autor para concretar una primera reunión y hablar sobre el tema para definir qué es lo que se quiere hacer.



**Cuarta fase: primera reunión** En esta reunión se hizo un repaso sobre cuál queríamos que fuese el resultado final. De esta manera se vieron dos problemas a solventar, el primero, encontrar los objetos, y el segundo, controlarlos de alguna manera.

**Quinta fase: investigación sobre tecnologías** Desde septiembre de 2013 hasta noviembre de ese mismo año se valoraron las diferentes opciones. Lo que quedó claro es que iba a ser en Android debida a que su código está en abierto.

**Sexta fase: aprendizaje Android** Para el aprendizaje Android se hizo llegar al autor una Tablet Galaxy, con sistema operativo Android 4.1, con la que poder iniciarse en la programación Android. Ya que se trataba de un concepto que no se había estudiado a lo largo de la carrera, necesitaba aprenderlo antes de todo. A modo de práctica se desarrollaron varias aplicaciones sencillas como un conversor digital o un contador de letras sobre un cuadro de texto.

Una vez adquirida cierta destreza en ejercicios sencillos, se pasó a intentar desarrollar un programa para encontrar objetos y reconocerlos.

**Septima fase: posicionamiento indoor** Ya que se trataba de un campo que no había tratado a lo largo de mis estudios valoramos cada una de los caminos posibles, acotando las posibilidades en dos opciones claras: modelado del entorno o reconocimiento visual.

El reconocimiento visual se dejó de lado ya que parecía ser algo más complicado que la primera opción, de manera que se puso en marcha la investigación y desarrollo de una aplicación que modelase una habitación de base rectangular basándose en las señales WiFi de diferentes puntos de la sala.

Para llegar a ello, en primer lugar, se desarrolló una aplicación capaz de leer las señales WiFi, su intensidad y punto de acceso, guardarlo y comparar la situación del terminal con todos los guardados, resultando dar una posición de entre estas, es decir, un sistema de localización basado en señales WiFi a relativa pequeña escala (5-10 metros).

Las reuniones se iban sucediendo cada varias semanas, ya que se compaginaban con el estudio de las asignaturas finales, habiendo llegado ya a marzo con un programa que modelaba una habitación rectangular,

pero que no funcionaba muy correctamente debido a la fluctuación de las señales de las que dependía.

**Octava fase: cambio de dirección** Después de varias reuniones y mucha investigación sobre el posicionamiento indoor sobre el que nos queríamos basar para encontrar el resto de objetos, quedó claro que era mejor optar por la otra posibilidad. De esta manera, en abril se enfocó el trabajo por esta otra vía, el reconocimiento de objetos. El problema era la llegada inminente de los exámenes finales, con lo que el desarrollo se postpuso para después de éstos.

**Novena fase: reconocimiento de objetos** El 15 de julio de 2014 se llevó a cabo una reunión en la que quedaba claro cuál iba a ser el resultado final: un reconocedor de marcadores que, a través de un interfaz de usuario sencillo, se pudiera actuar sobre objetos de la vida cotidiana, que serían representados en un mundo virtual. Pero la disponibilidad en verano fue escasa debido a empleos varios, así que no sería hasta agosto cuando se comenzase a trabajar con este objetivo más concreto.

**Décima fase: Desarrollo de la solución** A comienzos de agosto comenzaron las horas frente el ordenador, empezando por plantear un diseño y una arquitectura del programa, pensando en los requerimientos y recursos necesarios para llevarlo a cabo. En esta memoria se refleja este resultado.

## 5.2. Presupuesto

A continuación se justifican los costes globales de la realización del presente Trabajo Fin de Grado.

Desde la idea primigenia, hasta la entrega de esta memoria, pasan 20 meses, compatibilizándose con los estudios de los dos últimos cursos de la carrera, con exámenes finales, tareas del curso académico, así como otras obligaciones laborales de otro ámbito.

El presupuesto se basa en el desarrollo del proyecto del apartado 1.4:

1. Fase 1: Análisis del problema inicial: Prolongándose hasta 12 meses, con un total de unas 100 horas.
2. Fase 2: Aprendizaje inicial de Android: 50 horas.

3. Fase 3: Desarrollo de un primer sistema de localización indoor: 60 horas.
4. Fase 4: Concreción del problema final a solventar dividido en:
  - Fase 4.1: Concreción de las partes necesarias para resolver el problema: 5 horas.
  - Fase 4.2: Diseño del control domótico: 5 horas.
  - Fase 4.3: Diseño del sistema de realidad aumentada: 5 horas.
5. Fase 5: Desarrollo de un programa de uso sencillo para el manejo de la base de datos: 6 horas.
6. Fase 6: Desarrollo de un servidor ocupado en resolver las consultas del cliente, así como manejar la información de los diferentes ficheros: 15 horas.
7. Fase 7: Desarrollo de un cliente que realiza las consultas e interpreta las respuestas del servidor: 5 horas.
8. Fase 8: Desarrollo de un interfaz en el servidor que muestre los cambios que se realizan en el entorno: 3 horas.
9. Fase 9: Desarrollo de un interfaz en el cliente que muestre los objetos con los que se puede interactuar, así como sus diferentes acciones posibles: 4 horas.
10. Fase 10: Pruebas Cliente - Servidor: respecto a la petición y resolución de respuestas: 5 horas.
11. Fase 11: Desarrollo de un motor de reconocimiento de etiquetas: 10 horas.
12. Fase 12: Pruebas de reconocimiento a través de una aplicación Android que pueda leer e interpretar las diferentes etiquetas: 4 horas.
13. Fase 13: Aplicación final, de uso sencillo e intuitivo, que auna las partes anteriormente diferenciadas: 2 horas.
14. Fase 14: Pruebas finales realizadas para depuración del programa, con diferentes individuos no necesariamente familiarizados con el tema: 10 horas.

| Fase | Tema  | Número de horas |
|------|---|-----------------|
| 1    | Análisis del problema inicial                                 | 100             |
| 2    | Aprendizaje inicial de Android                                | 50              |
| 3    | Desarrollo de un primer sistema de localización indoor        | 60              |
| 4.1  | Concreción de las partes necesarias para resolver el problema | 5               |
| 4.2  | Diseño del control domótico                                   | 5               |
| 4.3  | Diseño del sistema de realidad aumentada                      | 5               |
| 5    | Desarrollo de un programa para el manejo de la base de datos  | 6               |
| 6    | Desarrollo de un servidor                                     | 15              |
| 7    | Desarrollo de un cliente                                      | 5               |
| 8    | Desarrollo de un interfaz en el servidor                      | 3               |
| 9    | Desarrollo de un interfaz en el cliente                       | 4               |
| 10   | Pruebas Cliente - Servidor                                    | 5               |
| 11   | Desarrollo de un motor de reconocimiento de etiquetas         | 10              |
| 12   | Pruebas de reconocimiento                                     | 4               |
| 13   | Aplicación final  | 2               |
| 14   | Pruebas finales   | 10              |
|      | <b>Total horas</b>  | <b>289</b>      |

Cuadro 5.1: Fases del proyecto y horas invertidas.

En la tabla 5.1 se puede ver la división del trabajo, concluyendo en un total de 289 horas. Estableciendo una tarifa de 20 euros a la hora, el coste de personal asciende a 5780 euros.

Por otro lado, hay que atender al equipo y programas que se han usado:

- Software

1. Eclipse (Windows). Con bibliotecas:
  - Android
  - AndAR
  - gson-2.3
2. Servidor Apache Tomcat (Linux).
3. MySQL
4. Scribus (programa de diseño de imágenes usado para editar marcadores)
5. Java (para el control del servidor y la base de datos)

- Hardware

1. Ordenador portátil con S.O. Windows.

- 2. Ordenador portátil con S.O. Linux.
- 3. Terminal Sony Experia Z2 con S.O. Android 4.4.
- Otros
  - 1. Papel.
  - 2. Bolígrafos.

| Concepto                            | Precio en euros |
|-------------------------------------|-----------------|
| Eclipse                             | 0               |
| Librería Android                    | 0               |
| Librería AndAR                      | 0               |
| Librería Gson-2.3                   | 0               |
| Servidor Apache Tomcat              | 0               |
| Librería MySQL                      | 0               |
| Programa Scribus                    | 0               |
| Java                                | 0               |
| Ordenador portátil con S.O. Windows | 780             |
| Ordenador portátil con S.O. Linux   | 300             |
| Terminal Sony Experia Z2            | 360             |
| Papel                               | 4               |
| Bolígrafos                          | 4               |
| <b>Total</b>                        | <b>1448</b>     |

Cuadro 5.2: Presupuesto de SoftWare y HardWare.

En la tabla 5.2 se tiene en cuenta el coste del SoftWare y del HardWare, que, junto a la tabla 5.1, componen la tabla 5.3, que muestra el presupuesto total, aplicando un añadido de costes indirectos de un 20 % y un I.V.A. del 21 %:

| Concepto                 | Precio en euros |
|--------------------------|-----------------|
| Coste personal           | 5780            |
| Coste SW y HW            | 1448            |
| Costes indirectos (20 %) | 1445.6          |
| I.V.A. (21 %)            | 1821.46         |
| <b>Total</b>             | <b>10495.06</b> |

Cuadro 5.3: Presupuesto total.

Ascendiendo a un total de diez mil cuatrocientos noventa y cinco euros y seis céntimos.

### 5.3. Planificación

En la tabla 5.4 se muestra una descomposición del trabajo en tareas de cada una de las fases generales citadas en el presupuesto, así como las horas que le corresponden a cada una de ellas.

La planificación del proyecto se ha previsto teniendo en cuenta las circunstancias personales a lo largo del proceso llegando a sumar 289 horas, lo que en términos de jornada laboral de 6 horas diarias, nos lleva a un total de 10 semanas, dos meses y medio, un tiempo cercano al previsto de 8 semanas. Es decir, el proyecto ha llevado un 25 % más de tiempo previsto.

En el apartado 5.5 se describe, a través de un diagrama de Gantt, la consecución de las diferentes tareas.

### 5.4. Reflexiones personales

Para la realización de este proyecto, ha sido necesario el estudio de varias tecnologías y plataformas de las cuales no se tenía ninguna noción, como Android y sus bibliotecas o la Realidad Aumentada, lo que ha supuesto un gran esfuerzo personal para adquirir estos conocimientos, poder conocerlos de forma individual, para más tarde unirlos con un fin común.

Al comienzo del proyecto, es decir, en el momento de encontrar documentación sobre las tecnologías y herramientas implicadas, la estimación del tiempo era difícil de calcular. Por esa misma razón se comenzó, con el curso académico, a investigar sobre este proyecto. Según pasaban las semanas se empezaba a atisbar la complejidad real de cada sistema, haciendo que el proponer reuniones e hitos fuera algo totalmente inservible, ya que se pedendía del ritmo de lectura y comprensión, haciendo que el proyecto se demorara.

Por otro lado, la biblioteca AndAR no cuenta con una documentación excesiva, ni con un API[50] que explique su uso, de manera que la única manera de poder desarrollar y avanzar algo sobre esta tecnología era la consulta a terceros, en foros sobre el tema que, naturalmente, usaban el inglés. Inter-

actuar en este idioma, así como leer documentación técnica en el mismo, ha supuesto una oportunidad para aprender esos tecnicismos que en el futuro hará falta saber y, sobre todo, para poder practicar y desenvolverme.

Como conclusión final, decir que, pese a la dificultad que en primera instancia se sabía que iba a tener este proyecto, desde el inicio se planteó como un reto al cual nos sumamos tanto mi tutor como yo mismo, con ganas de saber hasta dónde podemos llegar y de qué cosas podemos hacer con estos dos campos, tan interesantes y diferenciados, como son la Domótica y la Realidad Aumentada. Sin embargo, según se acercaba el final, se veía crecer esta aplicación, resultando ser precisamente aquello que en un principio nos habíamos propuesto. ¿Qué decir de poder llevar un sistema propio en tu Smartphone y poder hacer uso de él, aunque sea en un hogar virtual?. Espero poder continuar con este trabajo para llegar a esa propuesta del apartado 6.3, a través de estudios futuros, y poder desarrollar algo real, útil y en beneficio del conocimiento tecnológico.

## 5.5. Diagrama de Gantt

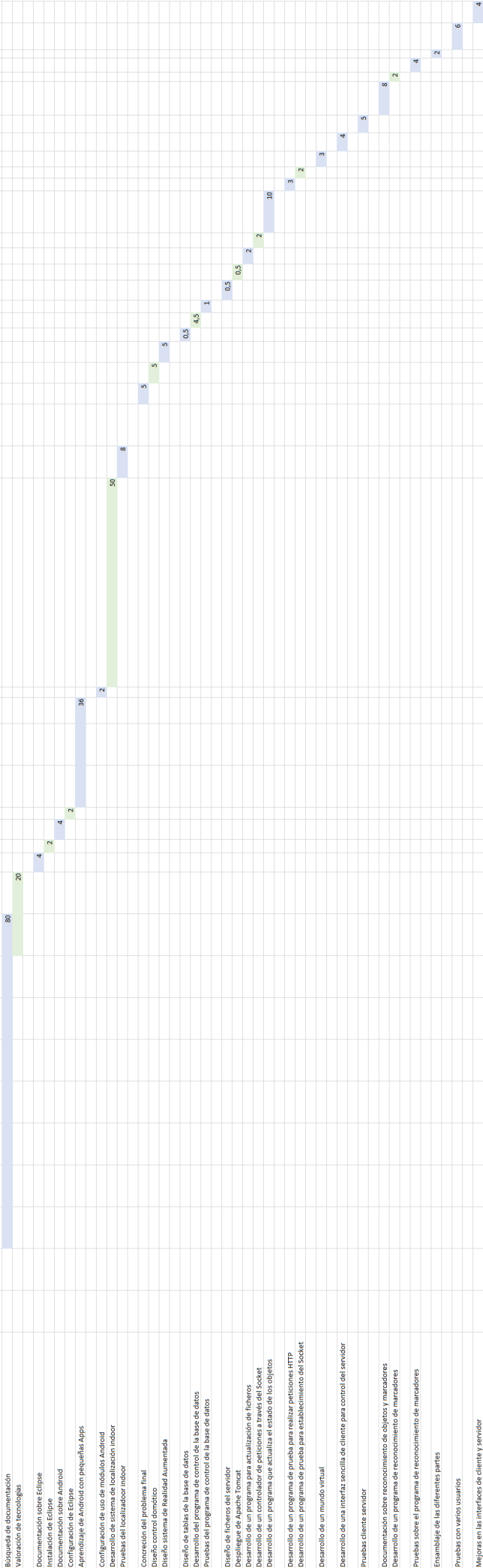
En la página 70 se puede ver el diagrama de Gantt.

| Fase      | Tema   | Número de horas |
|-----------|--|-----------------|
| <b>1</b>  | <b>Análisis del problema inicial</b>   | <b>100</b>      |
| 1.1       | Búsqueda de documentación de las diferentes tecnologías                        | 20              |
| 1.2       | Valoración de las diferentes tecnologías                                       | 80              |
| <b>2</b>  | <b>Aprendizaje inicial de Android</b>  | <b>50</b>       |
| 2.1       | Documentación sobre Eclipse  | 4               |
| 2.2       | Instalación de Eclipse   | 2               |
| 2.3       | Documentación sobre programación en Android                                    | 4               |
| 2.4       | Configuración de Eclipse con librería Android                                  | 2               |
| 2.5       | Desarrollo de pequeñas aplicaciones  | 36              |
| <b>3</b>  | <b>Desarrollo de un primer sistema de localización indoor</b>                  | <b>60</b>       |
| 3.1       | Configuración del programa para el uso del módulos como WiFi del terminal      | 2               |
| 3.2       | Desarrollo del programa  | 50              |
| 3.3       | Pruebas del programa   | 8               |
| <b>4</b>  | <b>Concreción del problema final</b>   | <b>15</b>       |
| 4.1       | Concreción de las partes necesarias para resolver el problema                  | 5               |
| 4.2       | Diseño del control doméstico   | 5               |
| 4.3       | Diseño del sistema de realidad aumentada                                       | 5               |
| <b>5</b>  | <b>Desarrollo de un programa para el manejo de la base de datos</b>            | <b>6</b>        |
| 5.1       | Diseño de las tablas   | 0.5             |
| 5.2       | Diseño del programa de control de la base de datos                             | 4.5             |
| 5.3       | Pruebas del programa de control de la base de datos                            | 1               |
| <b>6</b>  | <b>Desarrollo de un servidor</b>   | <b>15</b>       |
| 6.1       | Diseño de los ficheros   | 0.5             |
| 6.2       | Despliegue del servidor Apache Tomcat  | 0.5             |
| 6.3       | Desarrollo de un programa que modifique los archivos del servidor Apache       | 2               |
| 6.4       | Desarrollo de un controlador de peticiones de Sockets                          | 2               |
| 6.5       | Desarrollo de un programa que controle el estado de los objetos                | 10              |
| <b>7</b>  | <b>Desarrollo de un cliente</b>  | <b>5</b>        |
| 7.1       | Desarrollo de un programa de prueba para hacer peticiones HTTP                 | 3               |
| 7.2       | Desarrollo de un programa de prueba para realizar comunicaciones por Sockets   | 2               |
| <b>8</b>  | <b>Desarrollo de un interfaz en el servidor</b>                                | <b>3</b>        |
| 8.1       | Desarrollo de un mundo virtual que simule los cambios de estado de los objetos | 3               |
| <b>9</b>  | <b>Desarrollo de un interfaz en el cliente</b>                                 | <b>4</b>        |
| 9.1       | Desarrollo de un interfaz sencillo para realizar pruebas sobre lo anterior     | 4               |
| <b>10</b> | <b>Pruebas Cliente - Servidor</b>  | <b>5</b>        |
| 10.1      | Pruebas con varios usuarios  | 5               |
| <b>11</b> | <b>Desarrollo de un motor de reconocimiento de etiquetas</b>                   | <b>10</b>       |
| 11.1      | Documentación sobre reconocimiento de objetos y etiquetas                      | 8               |
| 11.2      | Desarrollo de un programa de reconocimiento de etiquetas                       | 2               |
| <b>12</b> | <b>Pruebas de reconocimiento</b>   | <b>4</b>        |
| 12.1      | Pruebas sobre el programa de reconocimiento                                    | 4               |
| <b>13</b> | <b>Aplicación final</b>  | <b>2</b>        |
| 13.1      | Ensamblaje de las diferentes partes del proyecto                               | 2               |
| <b>14</b> | <b>Pruebas finales</b>   | <b>10</b>       |
| 14.1      | Pruebas de todo el sistema con varios usuarios                                 | 6               |
| 14.2      | Mejoras sobre interfaces tanto cliente como servidor.                          | 4               |
|           | <b>Total horas</b>   | <b>289</b>      |

Cuadro 5.4: Planificación de tareas.



Tarea



# Capítulo 6

## Conclusiones y trabajo futuro

En este capítulo se hace un repaso de los objetivos planteados y el grado de satisfacción, además de una vista hacia el futuro, tanto de las tecnologías estudiadas para el desarrollo de este trabajo, como de este sistema.

### 6.1. Consecución de objetivos

En el apartado 1.3 se dispusieron una serie de objetivos, planteados como meta a la hora de desarrollar este proyecto. Veamos a continuación cómo se han ido superando:

- **Objetivos generales:**

1. **Implementar un sistema cómodo para interactuar con el entorno del hogar:** Se trata de una aplicación que se lleva en el terminal móvil, con lo que su uso es tan cómodo como el disponer del terminal al alcance de la mano, dónde todos los objetos pueden ser controlados con un único aparato.
2. **Desarrollar un sistema de control sobre los objetos que sea ampliable de forma sencilla y fácil de implementar:** En este sentido, se ha conseguido desarrollar un sistema completo cuya ampliación es bastante sencilla, ya que hace falta cambiar pocos parámetros y añadir pocas líneas de código en el momento en el que se quiera añadir más objetos para ser controlados.
3. **Desarrollar un sistema de identificación y reconocimiento de objetos simple:** En este proyecto la detección de los objetos

se ha basado en la detección de los marcadores que los representan. En este sentido hubiera sido mucho más atractivo haber hecho este reconocimiento directamente sobre el objeto. No obstante, se ha conseguido poder identificarlos e interactuar con ellos.

4. **Desarrollar un interfaz de usuario intuitivo y de fácil manejo para el control del sistema:** Desde luego el uso de la cámara e interactuar directamente sobre la visión a través de ella es el sistema más intuitivo, mucho más allá que un control del estilo “mando a distancia”, donde tener que saber cómo manejar un menú y recordar dónde se encuentra cada elemento dentro de ese menú.

- **Objetivos específicos:**

1. **Conocer el sistema Android y el uso de la biblioteca AndAR:** Sin lugar a dudas se ha conseguido conocer, en la profundidad que marcaba este proyecto, el sistema Android, así como la programación en esta plataforma. El uso de la biblioteca AndAR queda en contrapartida como algo no tan explorado, ya que no se usa su motor de dibujo de figuras geométricas.
2. **Integrar funcionalidades de Realidad Aumentada proporcionadas por Google en una aplicación Android:** El punto fuerte de esta aplicación es el de añadir a la realidad que vemos a través de la cámara una realidad creada a criterio del autor, cosa que desde luego se ha conseguido al superponer botones de control sobre la vista de la cámara.
3. **Desarrollar una base de datos de fácil manejo y uso, así como un sistema de ficheros donde guardar la información de estado del entorno del hogar:** El sistema de ficheros y la base de datos son de uso sencillo debido al desarrollo de un pequeño programa para poder manejar, cambiar, eliminar y añadir objetos.

Gracias a la consecución de todos estos objetivos se ha llegado al fin último, una aplicación que muestre las posibilidades que la realidad aumentada tiene en el control del hogar.

En definitiva se ha logrado una aplicación atractiva y novedosa para dispositivos Android con una interfaz de usuario sencilla y agradable que, mediante el uso de Realidad Aumentada se vuelve intuitiva, cumple con el objetivo de controlar el estado de diferentes elementos de un entorno.

## 6.2. Reflexión personal sobre el futuro

En el futuro nuestras mascotas estarán conectadas a Internet, nuestra báscula a nuestro médico de familia y un osito de peluche podrá regañar a nuestros hijos o incluso avisarnos si se han conectado a Youtube en mitad de la noche.

Podremos configurar notificaciones como “su asado está preparado” o “su hijo acaba de aparcar en su garaje”. Todo esto dependerá de lo que queramos que esté conectado, y cómo lo queramos.

A lo largo de este documento hemos tratado diferentes opciones para hacer funcionar un control sobre nuestros hogares y las opciones que nos da la domótica y la realidad aumentada para hacer de nuestra vida algo más sencillo. Sin embargo, todavía no hemos mirado hacia el futuro más o menos inmediato. En este capítulo veremos que opciones podremos tener dentro de pocos años.

### 6.2.1. Reconocimiento de objetos

En este trabajo se ha mencionado que algunos sistemas de A.R. trabajan con imágenes. Para ello hace falta un sistema que compare la imagen vista con una gran base de datos de imágenes.

Cada año Imagenet[51], desde 2010, propone una competición en la que se consiga mejorar de forma importante el reconocimiento de objetos, proponiendo que los participantes lleven a cabo un software para tal fin. El reto se basa en tres objetivos: detección, clasificación, y localización. El pasado 5 de septiembre de 2014 en el blog de investigación de Google[52], Christian Szegedy, ingeniero de Software de Google, posteo un artículo titulado “Construyendo un entendimiento mejor de las imágenes” en el que se describe como el equipo de google, GoogLeNet, ha ganado el primer premio de este año. La clave de su éxito ha sido la simpleza de su tecnología. Cada repaso de la imagen se asegura de forma más fina y ajustada sobre qué es lo que está viendo.

Google ha puesto a disposición de los desarrolladores toda la documentación para replicar, usar y modificar al gusto. Con esta base cualquier sistema de interacción con imágenes y/o marcas, como el de este trabajo, podría ser potenciado, haciendo de su uso algo todavía más sencillo.

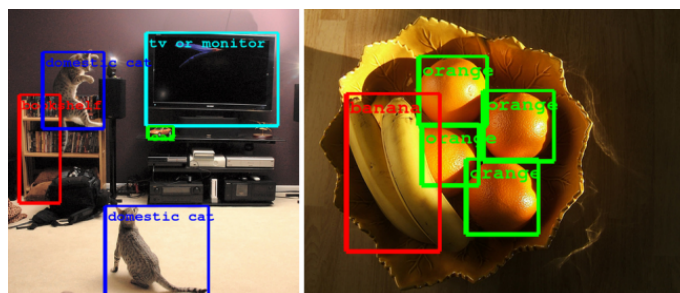


Figura 6.1: Objetos vistos con la solución de Google al reto ImageNet.

### 6.2.2. Realidad Aumentada

La realidad no sólo es la que conocemos, en un plano físico y terrenal. Cada vez estamos más cerca de un mundo dinámico y moldeable a nuestro parecer, de dar un nuevo sentido a lo que vemos, aumentando una realidad con contenidos más atractivos e interesantes para los usuarios.

Poco a poco se van desarrollando productos que formarán parte de nuestro kit de accesorios, como Google Glass[53]. Steven K. Feiner[54], profesor de la universidad de Columbia dijo que la integración de la RA a nuestras vidas “sería como mudarse de los audífonos grandes a los earbuds” añadiendo que “cuando son pequeños y cómodos no te sientes raro, sino cool”, idea que aplicable a la RA.

Distintos analistas tecnológicos comparten la opinión de que para el año 2020 la RA esté presente en nuestras vidas, aunque naturalmente existen diferentes opiniones sobre lo bueno o malo que será esta tecnología para nuestras vidas, basándose en la actividad social, en la conexión total del ser humano a redes y de posiblemente la desaparición de su intimidad.

Steve Mann, considerado el padre de la RA sostiene que será un beneficio su uso para la sociedad, aumentando la calidad de vida y la información disponible. Se trata de un sistema para ver mejor el mundo ya que hay muchas cosas en el mundo que no se pueden ver hasta el momento.

En contrapartida, Nicola Liberati[55], en la misma Augmented Reality Expo [56] mencionó que “no deberíamos poner nuestra atención únicamente en lo que podemos hacer con esta tecnología, sino más bien en lo que nos convertiremos por usarla”. Elementos como Google Glass nos puede hacer plantearnos la legalidad de alguna información que puede exponer su seguridad y confidencialidad así como sus efectos sobre los seres humanos, quizá haciéndonos más dependientes de máquinas.

En cualquier caso, ambos puntos de vista deben evolucionar observando cuáles son los efectos que esta tecnología provoca ya que estamos sólo en el

comienzo del cambio.



Figura 6.2: Ejemplo de uso futuro de R.A.

### 6.2.3. Casas inteligentes

La domótica ayuda a dar cierta inteligencia, cierta capacidad de decisión y automatismo a los hogares, convirtiéndolos en Casa Inteligentes. El cuidado de jardín o actividades típica del hogar que requieran cierto tiempo y sean bastante rutinarias se pueden sustituir por parte de las instalaciones haciéndolas por cuenta propia cuando sea configurado. Los dispositivos estarán conectados a una red, permitiendo su control mediante cualquier dispositivo con acceso a esa red.

Un ejemplo es el “proyecto Basalte: Altea Hills”. Se trata de un conjunto de residencias de lujo con tecnología domótica de última generación.

Tendremos capacidad de control sobre cualquier elemento que queramos pudiendo usar el mando a distancia de la TV, el móvil, la tablet o el smartwatch. Todo esto hará de nuestras casas un lugar más controlable, con mayor número de opciones que nos dejarán tiempo para dedicarnos a las actividades que realmente más nos interesen, aumentando nuestro tiempo libre.

Las casas producirán su propia electricidad y darán calor al mismo tiempo que reducen el consumo y la emisión de CO<sub>2</sub>. Las placas solares y la gestión de residuos serán pilares clave. Los interiores tendrán los mejores aislantes, pinturas que repelen el calor o el frío, serán adaptativas y cambiarán en función de los inquilinos. Los sistemas de higiene ayudarán a que los usuarios puedan mejorar su salud, conectándose incluso con su hospital. También monitorizarán su estado de salud y de ánimo pudiendo incluso cambiar el hilo musical.

Naturalmente los electrodomésticos inteligentes tendrán cada día mayor número de opciones, podremos interactuar con ellos con una sensación casi huma-

na, pudiendo pedir a nuestra cocina que nos prepare la comida con una orden hablada y contestada al mismo estilo de Siri[57].



Figura 6.3: Casa inteligente

#### 6.2.4. Internet de la cosas

Un mundo de objetos inteligentes interconectados entre sí.

Una de las ventajas de este mundo es que todos los sectores pueden sacar beneficio de estas tecnologías, siempre que haya conexión a una red, capacidad de análisis en tiempo real y protocolos que establezcan los procesos a seguir.

IDC[58] prevé que para el año 2020 el Internet de las Cosas generará unos ingresos de 7 billones de dólares a nivel mundial, que para entonces habrá 200 mil millones de dispositivos y objetos conectados a redes presentes en cualquier ámbito de nuestra vida.

Para que estos números ocurran se están enfocando los esfuerzos por parte de compañías como Intel en el desarrollo, entre otros, de procesadores de bajo consumo y servidores que den servicio sencillos, simples y fiables.

El gran volumen de datos que se estará intercambiando por segundo es un desafío al que hay que hacer frente, necesitando saber cómo y qué información básica y mínima se debe intercambiar. Pero no es el único. La idea del internet de las cosas es hacer que las actividades rutinarias sean automáticas y que las que no lo sean pasen a caracterizarse por su sencillez.

En los últimos meses los dispositivos conectados de uso personal como Smartw-

hatches o Gafas de AR y prendas tecnológicas, las wearables, son la novedad en el internet de las cosas: tienen en común que el usuario las lleva encima, como complemento, ofreciendo información en tiempo real.

Estamos hablando de que Internet llegue a todas las cosas. El internet de las cosas va mucho más allá de elementos electrónicos, quiere llegar a todos los objetos del planeta. Monitorizar cosechas, automatizar sistemas de riego dependiendo del estado de los alimentos, saber dónde está nuestro gato, informar del nivel de tinta que tiene nuestra pluma.

En una visión de futuro, todos aquellos que quieran estar conectados lo podrán estar. Naturalmente existen ciertas limitaciones como el acceso a Internet o la cobertura de datos, problemas sobre los cuales se trabaja de forma continua desde el nacimiento de las redes de datos.

Frederic Alluin, cofundador de Intelygenz (uno de los 2000 elegidos para formar parte del proyecto Google Glass), prefiere referirse a este término como “el Internet de Todas las Cosas”, ya que “básicamente dentro de poco todo estará conectado. Los objetos cotidianos y también los hábitos de las personas - si dormimos bien, mi salud en general - y cada uno de estos datos será almacenado, procesado y utilizado para mejorar nuestras vidas que, al fin y al cabo, es el objetivo final”.

El desarrollo de objetos inteligentes abre la vía a nuevas experiencias y formas de utilizar la tecnología en el día a día. Es seguir viviendo como hasta ahora pero con un acceso casi permanente a una información que nos hace la vida más sencilla e interesante, como LG Home Chat[59], que permite controlar los electrodomésticos e, incluso, programar pautas concretas para todos ellos con una sola orden.

Objetos como una cafetera que se conecta a Internet y, a través de una App, permite configurar el café a nuestro gusto son los denominados “Applios”, dispositivos que funcionan con una App y se controlan desde un terminal

Para septiembre de 2014 tiene previsto IABSpain[60] poner en marcha una comisión dedicada en exclusividad al Internet de las Cosas. Se trata de una empresa dedicada a la investigación de mercados, que en el caso del Internet de las Cosas, cree que son los usuarios los que van por delante, y que ellos son los que marcan el camino, debiendo las empresas identificar cómo generar valor añadido para satisfacer sus demandas.



Uno de los mayores problemas que existen en las telecomunicaciones y redes en general es el de la seguridad. En este mundo en el que todo está conectado un hacker malintencionado podría hacerse con el control de nuestras casas o vehículos, y hoy en día no estamos preparados para esto. Los sistemas de seguridad deben avanzar al mismo ritmo que estos sistemas antes de que lleguen a nuestras vidas cotidianas. Esta cuestión como su precio, que irá descendiendo de manera progresiva, son los dos obstáculos que no permiten todavía que esto sea una realidad de manera masiva, aunque su superación será cuestión de tiempo y dedicación. La pregunta es:

¿Estamos preparados para utilizar toda esta tecnología?

### **6.3. Líneas de trabajo futuro**

En esta sección se proponen diferentes formas de potenciar este trabajo, incluyendo nuevos sistemas que están en desarrollo. Si se dispusiera de más tiempo para poder ampliar, documentar y añadir este proyecto, se hubiera podido llegar a ese ideal, a esa idea primera que se planteó como semilla de este texto. Veamos a continuación que opciones podemos añadir.

#### **6.3.1. Reconocimiento de objetos**

Como se ha comentado en el apartado 6.2.1 el futuro prácticamente inmediato es el de poder realmente distinguir los objetos por sus características físicas, apuntando directamente con nuestra cámara del móvil en vez de tener que crear un sistema de reconocimiento de marcadores como se ha hecho en este proyecto.

De esta manera, incluir esta posibilidad en esta aplicación haría de ella algo más atractivo, sencillo e intuitivo de manejar, tanto para el usuario como para el desarrollador, que tendría que simplemente no preocuparse por realizar ni los marcadores ni una relación uno a uno con los objetos a los que representa.

#### **6.3.2. Realidad Aumentada**

En el apartado 2.2 se pueden ver las posibilidades de la realidad aumentada, que son bastante numerosas. A la hora de presentar la información sobre

la imagen que se ve a través de la cámara del terminal, se podría desarrollar una interfaz más atractiva:

- Añadir una barra de progreso para decidir subir y bajar la intensidad de las luces.
- En el caso de elementos que requieran indicar una posición, presentar un sistema animado para poder seleccionar la posición (como en persianas, puertas o ventanas)
- Con los elementos que se pueden reconocer pero que no se divisa su interfaz principal (como la pantalla de un ordenador que se pueda ver desde atrás), añadir una visualización en tiempo real, duplicando la imagen del ordenador sobre la pantalla del terminal.
- En aparatos de radio o de música añadir una presentación para controlar el volumen o el cambio de pista con los botones clásicos como Play, un triángulo; Pause, dos barras verticales; etc.
- Poder imprimir en impresoras en red.
- Reconocimiento de mascotas, pidiendo información de su estado, o cita con su veterinario.
- Etc.

### 6.3.3. Domótica: casas inteligentes

Con cada vez casas más inteligentes y autómatas, cabe la posibilidad de añadir un protocolo para añadir, no sólo las acciones que queramos nosotros sobre un elemento, sino reconocer e identificar las posibilidades propias del objeto, preconfiguradas por su fabricante, como hacer la compra por Internet por parte del refrigerador o encender la alarma del hogar.

### 6.3.4. Internet de las cosas

El máximo nivel al que podría llegar este trabajo es el de desarrollar un sistema de protocolos y lenguajes que estandaricen una casa inteligente, con un control domótico basado en el reconocimiento de objetos a través de la Realidad Aumentada. De esta manera, cada elemento estaría conectado a la red e intercambiaría información directamente con el terminal, de manera que este entendiera y pudiera mostrar al usuario las opciones que le corresponden a cada elemento sin necesidad de pasar dicha información por un servidor,

es decir, desde el objeto más pequeño al más grande del hogar, o incluso los seres vivos que de alguna manera (quizá una pulsera o un collar) estén conectados a la red, pudiera ser controlado desde nuestro terminal.



Figura 6.4: Habitación vista desde el terminal en un futuro.

# Capítulo 7

## Apéndices

En este apartado se incluyen los manuales de instalación del sistema así como de uso de la aplicación final.

### 7.1. Manual de instalación

Para la instalación hace falta un entorno Linux donde poder instalar MySQL, así como el servidor Apache y los programas de control del servidor.

- Instalar MySQL: Vaya a “<http://dev.mysql.com/doc/refman/5.0/es/linux-rpm.html>” y siga las instrucciones.
- Instalar Apache Tomcat: Vaya a “<http://www.nexolinux.com/apache-tomcat-7-instalacion/>” y siga las instrucciones.
- Crear el servidor: Copie la carpeta “FAR” en el directorio “webapps” de su servidor Apache.
- Instalar los .java: La carpeta javaFAR puede copiarla en cualquier lugar de su máquina, pero es necesario cambiar una línea de código. Vaya a la clase “CrearBaseFar.java” y verá como hay un atributo llamado “directorioCarpetasObjetos”, introduzca en ella el directorio donde ha instalado Apache, es decir “directorioApachewebappsFAR”

Para el uso de Eclipse y las librerías Android, vaya a la máquina donde tenga instalado Eclipse. Si no lo tuviera visite “<http://www.eclipse.org/downloads/>” y descarge la versión “Eclipse IDE for Java Developers”.

1. Descargar SDK de Android: Vaya a la siguiente dirección “<http://developer.android.com/intl/es/sdk/index.html>” y descargue la última opción disponible en cualquier ubicación.
2. Descargar el plugin Android para Eclipse: Haga click en “HelpInstall new software” y indique la URL de descarga “<https://dl-ssl.google.com/android/eclipse/>”. Seleccione el paquete “DDMS ” y “ Android Development Tools”
3. Configurar el plugin: En la ventana de configuración de Eclipse acceda a la sección de Android e indique en el cuadro “SDK location” la ruta del paso 1.
4. Configurar SDK targets: En la pestaña “WindowAndroid SDK and AVD Manager” vaya a la sección “Available Packages” y seleccione todos los paquetes deseados. En este caso es necesaria una versión superior a Android 4.2.

Una vez instalado el SDK de Android, importe el proyecto y siga las instrucciones del siguiente apartado, Manual de Usuario para poder disfrutar de la aplicación.

## 7.2. Manual de usuario

Una vez realizada la instalación correctamente (ver 7.1) se puede hacer uso de la aplicación. Para ello debe configurar previamente lo siguiente:

1. Lanzar el servidor: Para ello vaya a directorioInstalacionApacheapachetomcat-xxxbin/ (siendo x los número referentes a la versión instalada). Una vez en ese directorio ejecute “./startup.sh”. (Para detener el servidor, en este mismo directorio ejecute “./shutdown.sh”
2. Lanzar Mundo.java y el Menú de control de la base de datos: Para ello vaya a la carpeta directorioInstalacionjavaFAR y ejecute el siguiente comando: “javac -classpath .lib\*:.class:. MenuBaseDatos. De esta manera verá lo mismo que en la captura de la figura 4.11 así como aparecerá en una nueva ventana una vista del mundo que controlar, igual que en la figura 4.10
3. Conocer la dirección IP del servidor: Vaya al terminal de la máquina donde ha lanzado su servidor y escriba “sudo ifconfig”. Introduzca su contraseña si se requiere, y recuerde la dirección IP de la red a la que

está conectado (en caso de ser una VPN el terminal debe estar también conectado a la misma, si se trata de una dirección pública no hace falta que el terminal lo esté).

4. Cambiar la dirección IP en la aplicación: Arranque Eclipse y en la carpeta del proyecto vaya al directorio src y entre en la clase “CustomObject1.java”. Verá que uno de los atributos de esta clase se llama “serverAddress”, cambie la dirección que viene por defecto e introduzca la que anotó en el paso anterior.
5. Lanzar la aplicación en el terminal: Conecte su terminal al puerto USB. Haga click derecho sobre la carpeta que recoge todo el proyecto dentro de su navegador de proyectos de Eclipse. Vaya a “Run As” y haga click en “Android Application”. La aplicación quedará instalada en su terminal.
6. Usar la aplicación: Una vez haya seguido estos pasos tendrá, por un lado la vista del mundo virtual y, por otro, la aplicación corriendo en su terminal. Verá en la parte superior izquierda de la imagen un candado de color rojo que indica que la aplicación está bloqueada. Si mantiene pulsado durante un segundo en cualquier punto de la pantalla la desbloqueará, volviéndose el candado de color verde e identificando los marcadores del mundo virtual.



# Referencias

- [1] Es un sistema de posicionamiento en interiores (IPS indoor positioning system) que Apple Inc. denomina como “una nueva clase de transmisores de bajo consumo y bajo coste que pueden notificar a dispositivos iOS7 de su presencia por proximidad”.
- [2] Near field communication (NFC) es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.
- [3] RFID (siglas de Radio Frequency IDentification, en español identificación por radiofrecuencia) es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio.
- [4] Android es un sistema operativo basado en el kernel de Linux, diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, y también para relojes inteligentes, televisores y automóviles, inicialmente desarrollado por Android Inc., que Google respaldó económicamente y más tarde compró esta empresa en 2005.
- [5] AndAR es un proyecto que permite el uso de Realidad Aumentada en plataforma Android.
- [6] Asociación Española de Domótica e Inmótica (CEDOM): <http://www.cedom.es/es>
- [7] Specific Actions for Vigorous Energy Efficiency.
- [8] <http://www.picoelectronics.com/>
- [9] Plug-and-play o PnP (en español “enchufar y usar”) es la tecnología o cualquier avance que permite a un dispositivo informático ser conectado



a una computadora sin tener que configurar, mediante jumpers o software específico (no controladores) proporcionado por el fabricante, ni proporcionar parámetros a sus controladores.

- [10] El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado OSI (en inglés, Open System Interconnection “sistemas de interconexión abiertos”) es el modelo de red descriptivo, que fue creado por la Organización Internacional para la Estandarización (ISO) en el año 1980. Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones.
- [11] Carrier sense multiple access with collision avoidance. Protocolo de control de acceso al medio.
- [12] Spaced Frequency Shift Keying
- [13] Internet Protocol
- [14] <https://www.bsria.co.uk/>
- [15] La Norma UNE-EN 15232 ha sido elaborada por el comité técnico AEN/CTN 100 Climatización, de cuya secretaría se hace cargo la Asociación de Fabricantes de Equipos de Climatización (AFEC).
- [16] Instituto para la Diversificación y Ahorro de la Energía.
- [17] <https://www.layar.com/>
- [18] <http://www.junaio.com/>
- [19] <http://www.wikitude.com/>
- [20] Un código QR (quick response code, «código de respuesta rápida») es un módulo útil para almacenar información en una matriz de puntos o un código de barras bidimensional creado en 1994 por la compañía japonesa Denso Wave, subsidiaria de Toyota.
- [21] Un archivo con extensión .pat es un patrón, Pattern.
- [22] <http://www.aurasma.com/>
- [23] <http://daqri.com/>
- [24] <http://www.armedia.it/>

- [25] Sistema de Diagnóstico Médico y Cirugía Asistida por Ordenador, CAM-DASS en inglés..
- [26] <http://www.learnar.org/>
- [27] <http://www.tat.se/>
- [28] Más info. en <http://www.aplicacionesnokia.es/nokia-point-and-find-para-nokia-5800-n97/>
- [29] Se puede descargar en Google Play:  
<https://play.google.com/store/apps/details?id=com.loft.lookator2>
- [30] <http://www.androoid.net/3dcoche/>
- [31] <http://www.playar.es/>
- [32] <https://support.google.com/websearch/answer/166331>
- [33] Se puede descargar en Google Play:  
<https://play.google.com/store/apps/details?id=com.google.android.stardroid>
- [34] Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.
- [35] ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radio-difusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN).
- [36] Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.
- [37] Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.
- [38] Puede encontrar documentación en <http://tomcat.apache.org/>

- [39] JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.
- [40] MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.
- [41] Atómica, Consistente, Aislada y Duradera.
- [42] <https://source.android.com/>
- [43] Un kit de desarrollo de software o SDK (siglas en inglés de software development kit) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto.
- [44] Es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, éstos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web.
- [45] JavaServer Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos. JSP usa el lenguaje de programación Java.
- [46] Puede consultar su código en: <https://code.google.com/p/andar/source/browse/trunk/AndAR/src/edu/dhbw/andar/ARObject.java?r=177>
- [47] ARToolKit es una biblioteca que permite la creación de aplicaciones de realidad aumentada.
- [48] Un frame, fotograma o cuadro es una imagen particular.
- [49] Normalmente se usa para correr código en un hilo diferente.
- [50] Interfaz de programación de aplicaciones (IPA) o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.
- [51] <http://image-net.org/>.

- [52] <http://googleresearch.blogspot.com.es/>.
- [53] <https://www.google.com/glass/start>.
- [54] <http://www.cs.columbia.edu/~feiner/>
- [55] <http://augmentedworldexpo.com/people/nicola-liberati/>
- [56] <http://augmentedworldexpo.com/>
- [57] Siri es una aplicación con funciones de asistente personal para iOS. Esta aplicación utiliza procesamiento del lenguaje natural para responder preguntas, hacer recomendaciones y realizar acciones mediante la delegación de solicitudes hacia un conjunto de servicios web que ha ido aumentando con el tiempo.
- [58] International Data Corporation (IDC) es el principal proveedor mundial de inteligencia de mercado, servicios de consultoría y eventos para los mercados de tecnología de la información, telecomunicaciones y tecnología de consumo.
- [59] <http://www.lg.com/ae/press-release/lg-rolls-out-premium-smart-appliances-that-chat>
- [60] <http://www.iabspain.net/>